

USING SAS PC WITH WINDOWS 95, 98, 2000, and WINDOWS NT**Biometry & Statistics 302/602, Spring 2002****By Dr. Sophonie Nshinyabakobeje****ABSTRACT**

Over the past few years there has been significant computation improvement in the area of applied statistical analysis using the SAS (Statistical Analysis System) Program. The availability of faster and more powerful computers and newer operating systems have made it possible to analyze data more efficiently using PCs. The current SAS tutorial is based on WINDOWS 95, 98, 2000, XP and Windows NT. In the first part of this tutorial document instructions on data creation and manipulation, creation of new variables are demonstrated. Regression analysis is illustrated as an introduction for students taking BTRY302/602 as they have learned this topic in BTRY 261/601. This document is aimed at new users of SAS, former or current SAS users who are interested in the aforementioned operating systems. SAS Interactive is also used to get more insights into issues of residual diagnostic plots for assessing assumptions underlying simple linear regression analysis.

In the second part of this tutorial are commonly used statistical analysis methods in the context of general linear models, general linear mixed models, and generalized linear models. Other aspects of SAS judged important are also included in the second part of this tutorial manual.

Key Words: regression, SAS, contrasts, multiple comparisons, diagnostic plots, equal variance assumption, normality, PROC PLOT, PROC REG, PROC GLM, PROC MIXED, PROC LOGISTIC.

BU-1593-M in the Technical Report series of the Department of Biological Statistics and Computational Biology, Cornell University, Ithaca, New York 14853, Spring 2002.

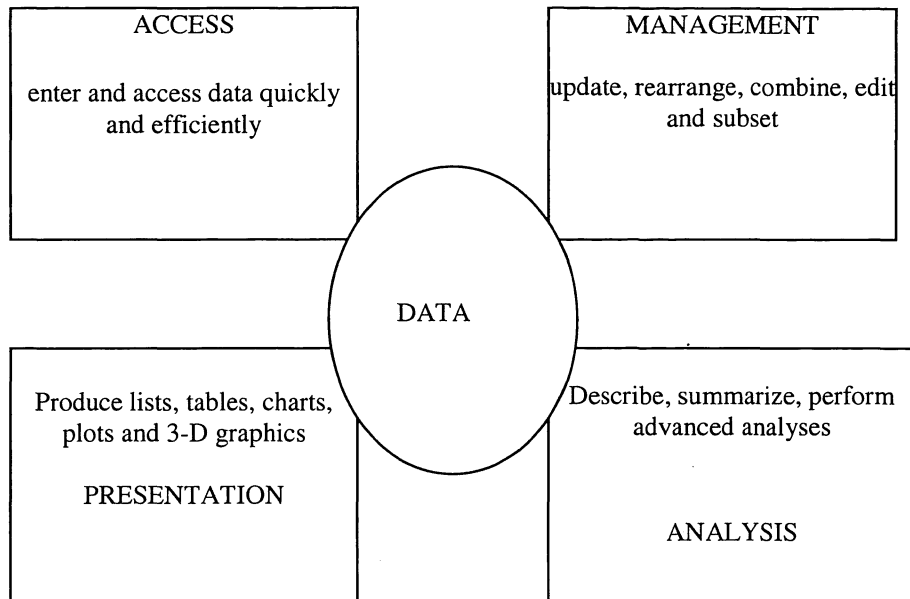
© 2002 by Dr. Nshinyabakobeje, S. All rights reserved.

TABLE OF CONTENTS

TABLE OF CONTENTS	1
PART I. SAS TUTORIAL MATERIAL.....	3
Overview Of The SAS System.....	3
Two Steps Needed In The SAS Programming Language.....	3
Characteristics Of A SAS Data Set	4
Creating A Data Set Using Notepad.....	5
Creating A SAS Program For Data Analysis	6
Using SAS Function Keys.....	11
Creating New Variables In SAS.....	13
Making Scatter Plots Using SAS Commands.....	15
Checking Errors In A SAS Program File	16
General Observations	18
Making Scatter Plots Using SAS Interactive.....	18
Making A Scatter Plot With Loess Smooth Using SAS Interactive.....	22
Parametric Method	22
Non-parametric Method	25
Performing Regression Analysis Using SAS	27
Creating Regression Diagnostic Plots Using SAS Interactive.....	30
Plot Of Residuals Versus Predicted Values.....	31
Plot Of Cook's Distance Versus Observation Numbers	32
Assessing The Normality Assumption	34
Assessing The Equal Variance Assumption.....	35
Producing A Report From A SAS Output	37
Recommended Reference.....	38
PART II. USEFUL INFORMATION ON SAS PROGRAM.....	39
Appendix 1: Data Management And Types Of SAS Data Sets.....	39
Appendix 2: Important SAS Commands For Old Homework.....	42
Appendix 3: Diagram Showing SAS Program's Potential Analyses.....	58

PART I. SAS TUTORIAL MATERIAL

Overview Of The SAS System



Source: Statistical Analysis System (SAS)

- The SAS system is a software system/package for data analysis. SAS provides tools for: information storage and file handling; data modification and management; statistical analysis; and report writing.
- The SAS system is an integrated application system that provides complete data processing and analysis capabilities.
- The SAS system is a powerful programming language and a collection of ready-to-use programs called procedures or PROC's, which can perform a large variety of applications (see Appendix 3).

Two Steps Needed In The SAS Programming Language

- The SAS language has its own vocabulary and syntax - words and the rules for putting them together. A SAS statement is a string of SAS keywords, SAS names, and special characters and operators *ending in a semicolon* that instructs SAS to perform an operation or gives SAS information. A sequence of SAS statements is called a SAS program.
- A SAS program consists of two kinds of steps: DATA steps and PROC steps. DATA and PROC steps can appear in any order, and any number of DATA and PROC steps can be used in a SAS program.
- Usually, DATA steps create SAS data sets, and PROC steps process SAS data sets.
- The DATA step can include statements telling SAS to create one or more new SAS data sets and programming statements that perform the manipulations necessary to build the data sets.
- The DATA step begins with a DATA statement and can include any number of program statements.

- You can use the DATA step for these purposes:
 - retrieval: getting input data;
 - editing: checking for errors in the data and correcting them; computing new variables;
 - outputting: write data sets to disk;
 - creating: producing new SAS data sets from existing ones by subsetting, merging, and updating.
- A DATA step is a group of SAS statements that begins with a DATA statement.

```
DATA ONE;
INFILE 'A:YIELD.DAT';
INPUT TREAT REP YIELD;
LOGY=LOG (YIELD);
RUN;
```

These SAS statements will be explained in forthcoming sections of this tutorial.

- The PROC step (or PROCEDURE step) instructs SAS to call a procedure from its library and to execute that procedure, usually with a SAS data set as input.
- The PROC step begins with a PROC statement. Other statements in the PROC step give the program more information about the results that you want.

Important Note: Every SAS statement ends with a semicolon. Otherwise, there is an error message or a misinterpretation by SAS! If your program does not run, recall the program editor and check first for missing semicolons.

Characteristics Of A SAS Data Set

The SAS system reads data (letters or numbers) in various forms and organizes them into a SAS data set. A SAS data set stores data in a form that the system can identify and manage as a unit. Once the data have been organized into a SAS data set, you can access, analyze, revise, and display the data.

The data consist of the following components: data value, variable, and observation.

- Data value is a single unit of information.
- Variable is a set of data values in each column.
- Observation is a set of data values in a row for all variables.

Variables				
NAME	SEX	AGE	HEIGHT	WEIGHT
Aubrey	M	41	74	170
Ron	M	42	68	166
Carl	M	32	70	155
Antonio	M	39	72	167
Deborah	F	30	66	124
Jacqueline	F	33	66	.
Helen	F	26	64	121
David	M	30	.	158
James	M	53	72	175
Michael	M	32	69	143
Ruth	F	47	69	139

Missing values

Data values

The data set above contains 5 variables, 11 observations, and 55 data values two of which are missing.

Now you are ready to create a data set file and a SAS program file.

Notepad is used to create our data set. You could also use **Word** to create the data set; in which case you should save the file as a text file. A SAS program can also be created so as to contain the data set but this option is **not** recommended. One can use a SAS program to read different data sets. The same SAS program can be slightly modified to make other statistical analyses of interest.

Creating A Data Set Using Notepad

The effect of manganese (Mn) on wheat growth is examined in the article "Manganese deficiency and toxicity effects on growth, development, and nutrient composition in wheat" (*Agronomy Journal*, 1984). Modified data for a random sample of 10 plants in this experiment are given in the table below. The response variable is the plant height (in mm) and the predictor variable is the amount of Mn added (in PPM).

HEIGHT = adjusted plant height (mm)	AMOUNT = amount of Mn
102.4	0.37
136.9	0.67
193.6	1.00
202.5	1.22
211.6	2.72
176.4	7.39
176.4	14.44
160.0	24.53
136.9	29.96
90.0	54.60

Source: Adapted from the *Agronomy Journal*, 1984.

Using the data above, we will first create a data set using a Text Editor. In this particular case, Notepad is used to create a data set called **PLANT.TXT**.

☐ **Start > Programs > Accessories > Notepad.**

Now, type the following data set in Notepad. **Please use only the spacebar to separate your data values.** If you use the tab key to separate your values, SAS will not read your data set and will give you an error message. So please, do not use the tab key.

```
102.4      0.37
136.9      0.67
193.6      1.00
202.5      1.22
211.6      2.72
176.4      7.39
176.4      14.44
160.0      24.53
136.9      29.96
90.0       54.60
```

The next step is to save the data set above on a diskette.

☐ Insert your diskette in drive A: (or drive B:)

☐ Save the file on a diskette in drive A: by choosing **File > Save >** Under File name type A:\PLANT > **Save**. The file is automatically given the extension **TEXT** by Notepad and has the name **PLANT.TXT**.

☐ **Choose File > Exit**. This is how you exit Notepad.

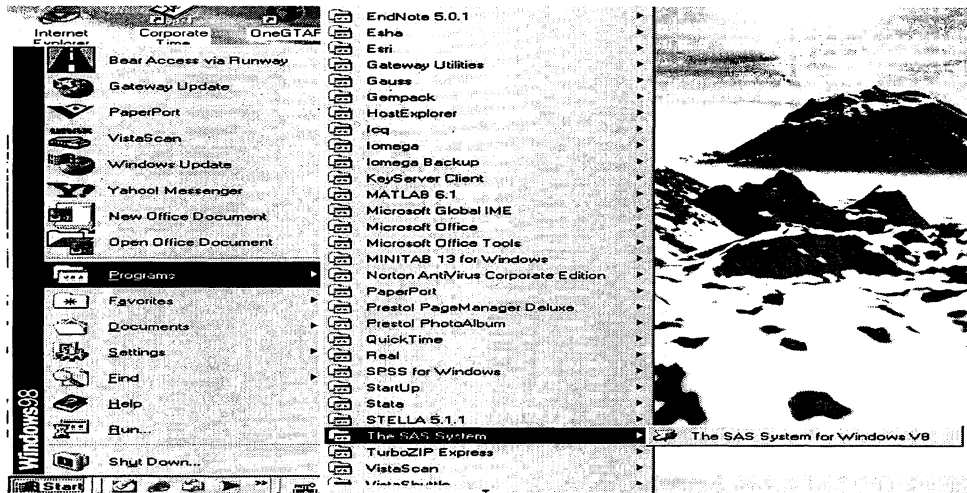
Note: The data set you just created has been saved on a diskette in drive A: and has the name **PLANT.TXT**. Now you are going to create the SAS Program for analyzing this data set.

Creating A SAS Program For Data Analysis

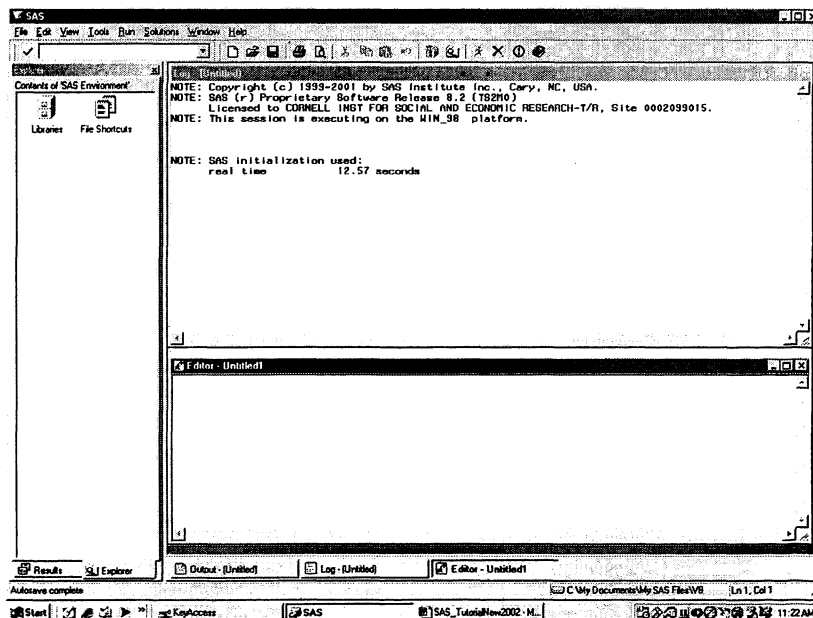
Start SAS Release 8.2 as follows:

❑ **Start > Programs > The SAS System > The SAS System for Windows V8**

The menu path for opening SAS is shown below.



After opening SAS, one is prompted to the following screen with three windows (see below). The vertical window at the left hand side is for Explorer and Results. Explorer contains SAS Libraries and Files short cuts. Results of SAS analysis are organized in a tree structure. The upper window at the right hand side is a Log Window containing SAS program executed statements and error messages if any. The bottom window is the Program Editor Window. Another window is the Output window the tab of which appears at the bottom windows' menu.



Please, create the following SAS program in the Program Editor Window above. SAS program is not case-sensitive so use either upper or lower case letters or both as you want. SAS commands are explained in bold face characters. So do not type information in bold face characters.

```
/*  
THIS PROGRAM IS USED TO CREATE A SMALL SAS PROGRAM.  
TOPICS OF INTEREST INCLUDE: DATA MANIPULATION,  
PLOTING DATA USING SAS INTERACTIVE, PLOTING DATA USING SAS  
COMMANDS, AND REGRESSION ANALYSIS.  
WRITTEN BY: LAST NAME, FIRST NAME OF STUDENT.  
DATE: JANUARY 2002.  
*/
```

A SAS comment is enclosed between symbols /* and */ or between * and ; symbols on each line.

```
OPTIONS LS=79 NOCENTER NODATE NONUMBER;
```

The OPTIONS statement tells SAS to limit the output lines to 79 characters (LS=79). LS stands for line size. The output will not be centered on each page (NOCENTER), and neither the date (NODATE) nor the page numbers (NONUMBER) will be displayed on the SAS output.

```
TITLE 'DATA MANIPULATION USING SAS PROGRAM.';  
TITLE2 'LAST NAME, FIRST NAME OF STUDENT. BTRY 602. DATE:';
```

These two TITLE lines will be printed on each page of the SAS output. You can add a title, or change titles anywhere in your SAS program. The TITLE statement is enclosed in single quotes ' '.

```
DATA TUTORIAL;
```

The DATA statement instructs SAS to create a temporary data set called WORK.TUTORIAL. A name of a data set starts with a letter and has at most 8 characters of numbers, letters, or underscores.

```
INFILE 'A:PLANT.TXT' ;
```

The INFILE statement tells SAS that the data are read from a file called PLANT.TXT on drive A:.

```
INPUT HEIGHT AMOUNT;
```

The INPUT statement tells SAS to name the two columns of data in WORK.TUTORIAL respectively, HEIGHT and AMOUNT. Each column in a data set must be given a name. A variable has at most 8 characters and must start with a letter.

```
LABEL HEIGHT ='ADJUSTED PLANT HEIGHT'  
      AMOUNT ='AMOUNT OF NUTRIENT' ;
```

The LABEL statement assigns labels to variables for easy reference. Variable HEIGHT is the adjusted plant height and AMOUNT is the amount of nutrient.

```
PROC PRINT DATA=TUTORIAL;
```

The PROC PRINT tells SAS to print the content of data set called WORK.TUTORIAL.

```
RUN;
```

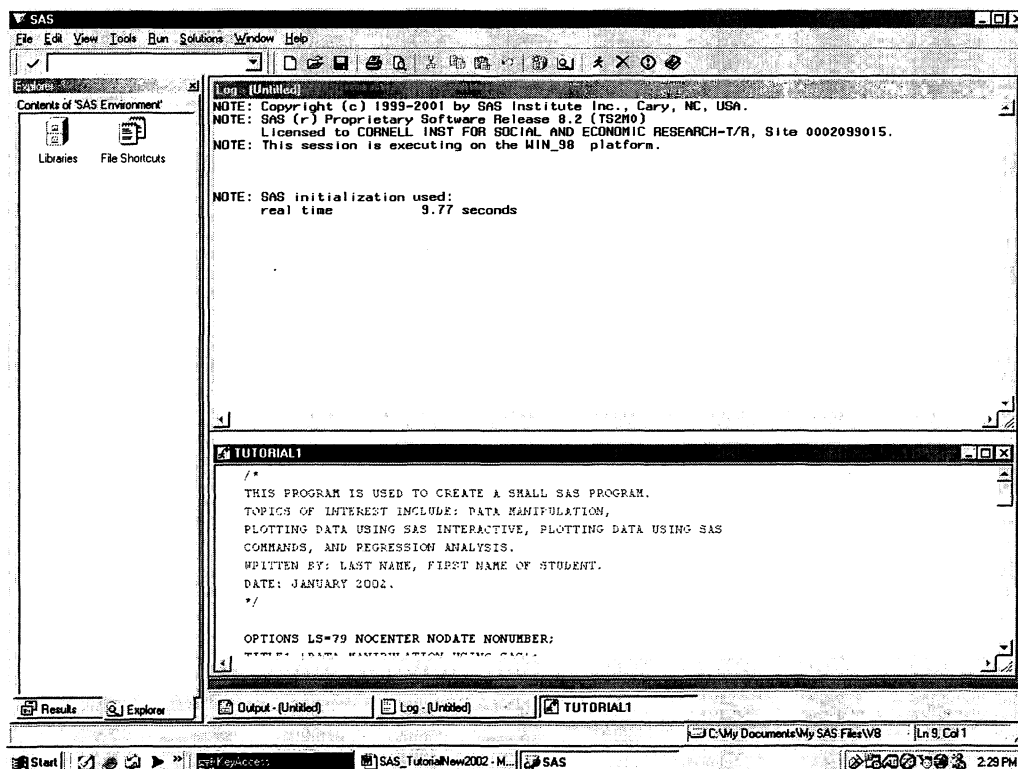
```
QUIT;
```

The RUN statement instructs SAS to process (execute) all statements and PROCs in a SAS program file. QUIT statement terminates a SAS program procedures without any further descriptor creation.

A summary of the SAS program you just created is provided below.

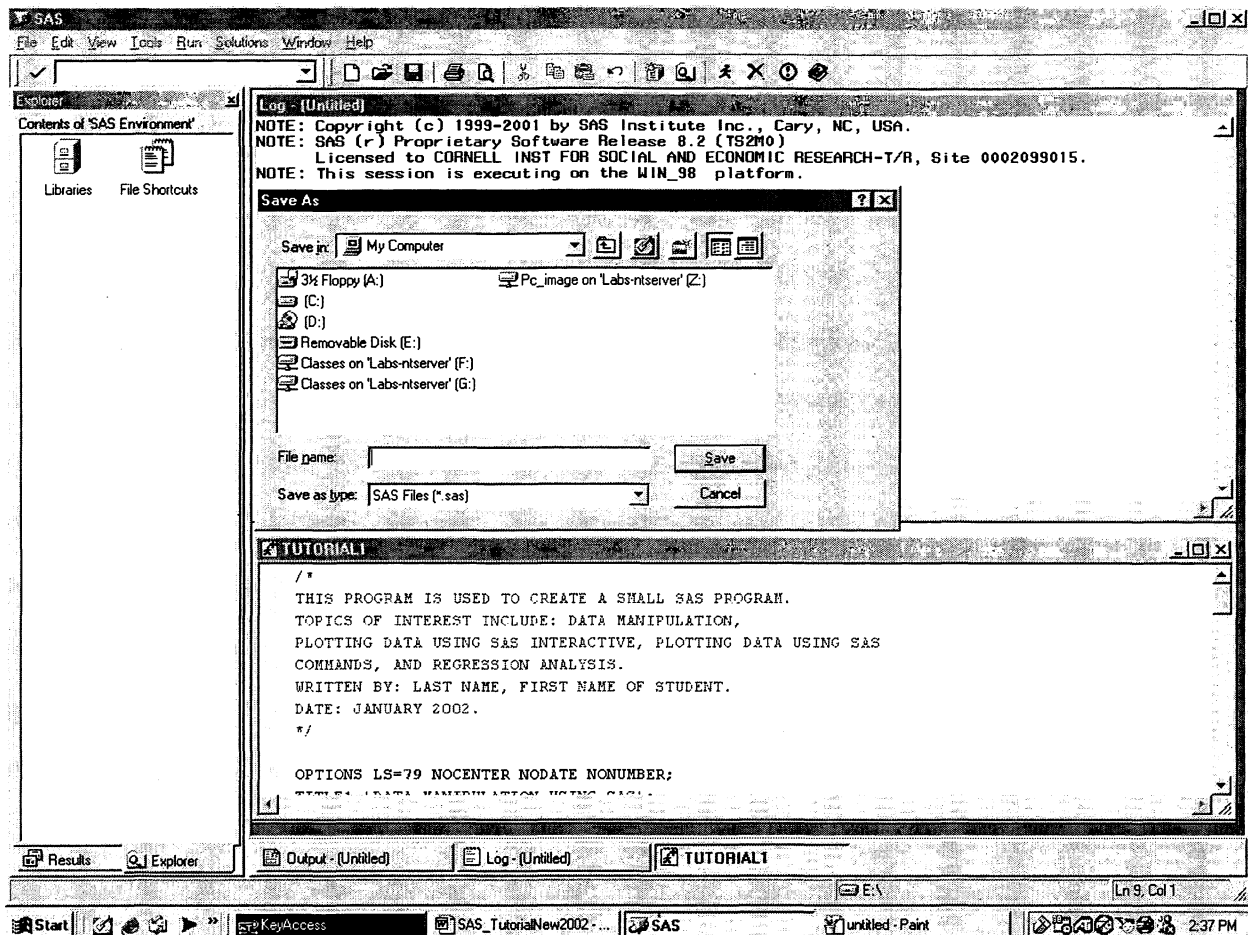
```
/*
THIS PROGRAM IS USED TO CREATE A SMALL SAS PROGRAM.
TOPICS OF INTEREST INCLUDE: DATA MANIPULATION,
PLOTTING DATA USING SAS INTERACTIVE, PLOTTING DATA USING SAS
COMMANDS, AND REGRESSION ANALYSIS.
WRITTEN BY: LAST NAME, FIRST NAME OF STUDENT.
DATE: JANUARY 2002.
*/
OPTIONS LS=79 NOCENTER NODATE NONUMBER;
TITLE 'DATA MANIPULATION USING SAS PROGRAM.';
TITLE2 'LAST NAME, FIRST NAME OF STUDENT. BTRY 602. DATE: ';
DATA TUTORIAL;
INFILE 'A:PLANT.TXT';
INPUT HEIGHT AMOUNT;
LABEL HEIGHT ='ADJUSTED PLANT HEIGHT'
      AMOUNT ='AMOUNT OF NUTRIENT';
PROC PRINT DATA=TUTORIAL;
RUN;
QUIT;
```

At this point, your SAS program should look as follows in the Program Editor Window.



- ☐ Save your SAS program above on your diskette in Drive A: and name the file TUTOR1.SAS.
- ☐ Choose File > Save.

You are prompted to the dialog box below.



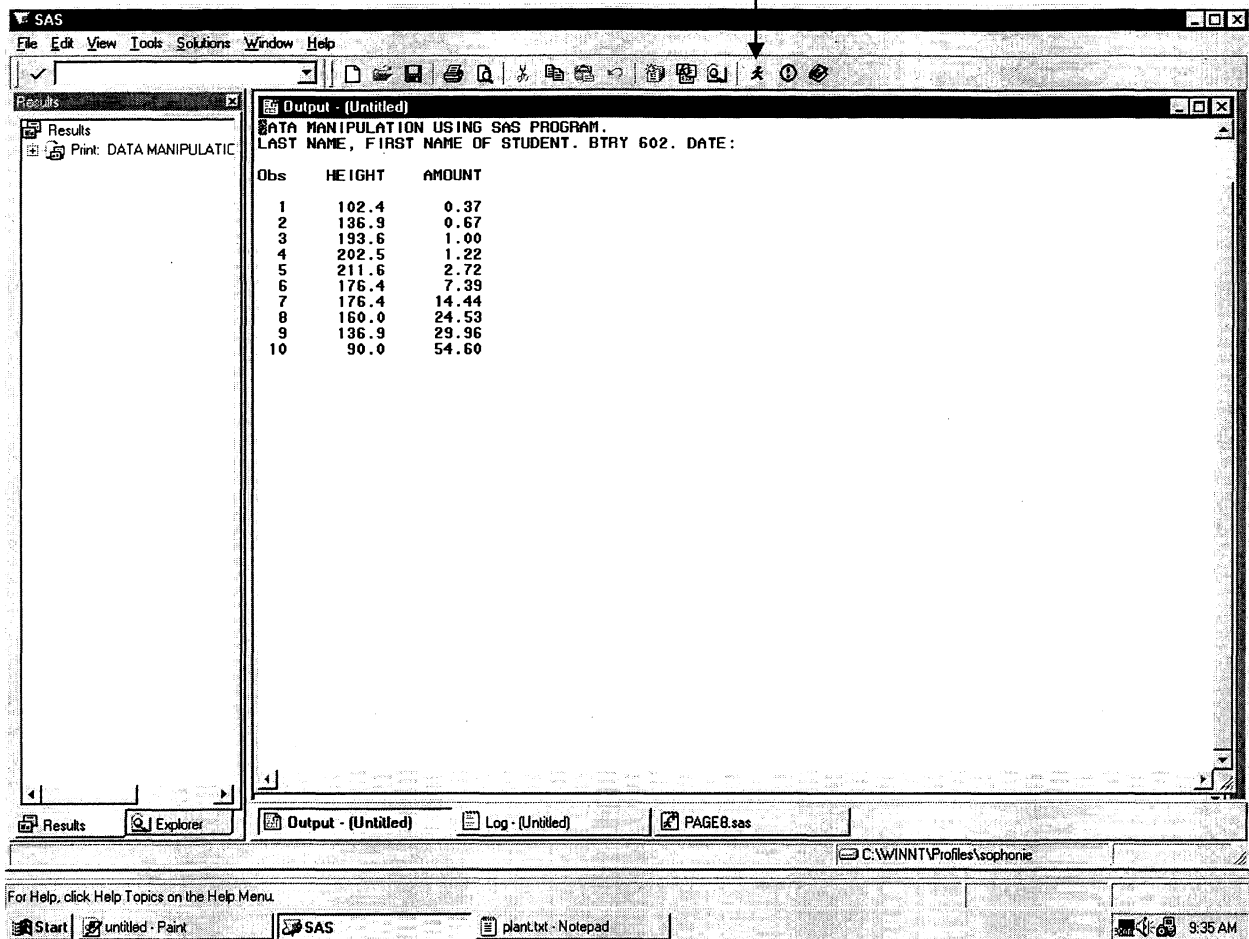
- ☐ Scroll under if necessary then choose $3\frac{1}{2}$ Floppy (A:)
- ☐ In the box labeled **File name**, type **TUTOR1** and click on **Save** tab. Your file has been saved on your diskette in Drive A: and has the name **TUTOR1.SAS**. SAS automatically assigns the extension SAS to the file name.
- ☐ Run the SAS program just created as follows: select **Run ► Submit**.

Two other ways of running SAS are as follows:

- ☐ You may also run SAS by clicking the SUBMIT icon with your mouse left button; or by pressing the key function F3 on the keyboard. The Submit icon is the third icon (showing a running person) from the right upper corner at the top of the dialog box above. The submit icon on the diagram below (see arrow).
- ☐ Another handy way to run SAS is to highlight the portion of interest using the left-mouse button, followed by a click of the run icon. This is helpful when the SAS program file is very long and one wants to run part of it.

Try to run SAS using each the three methods above.

Submit icon



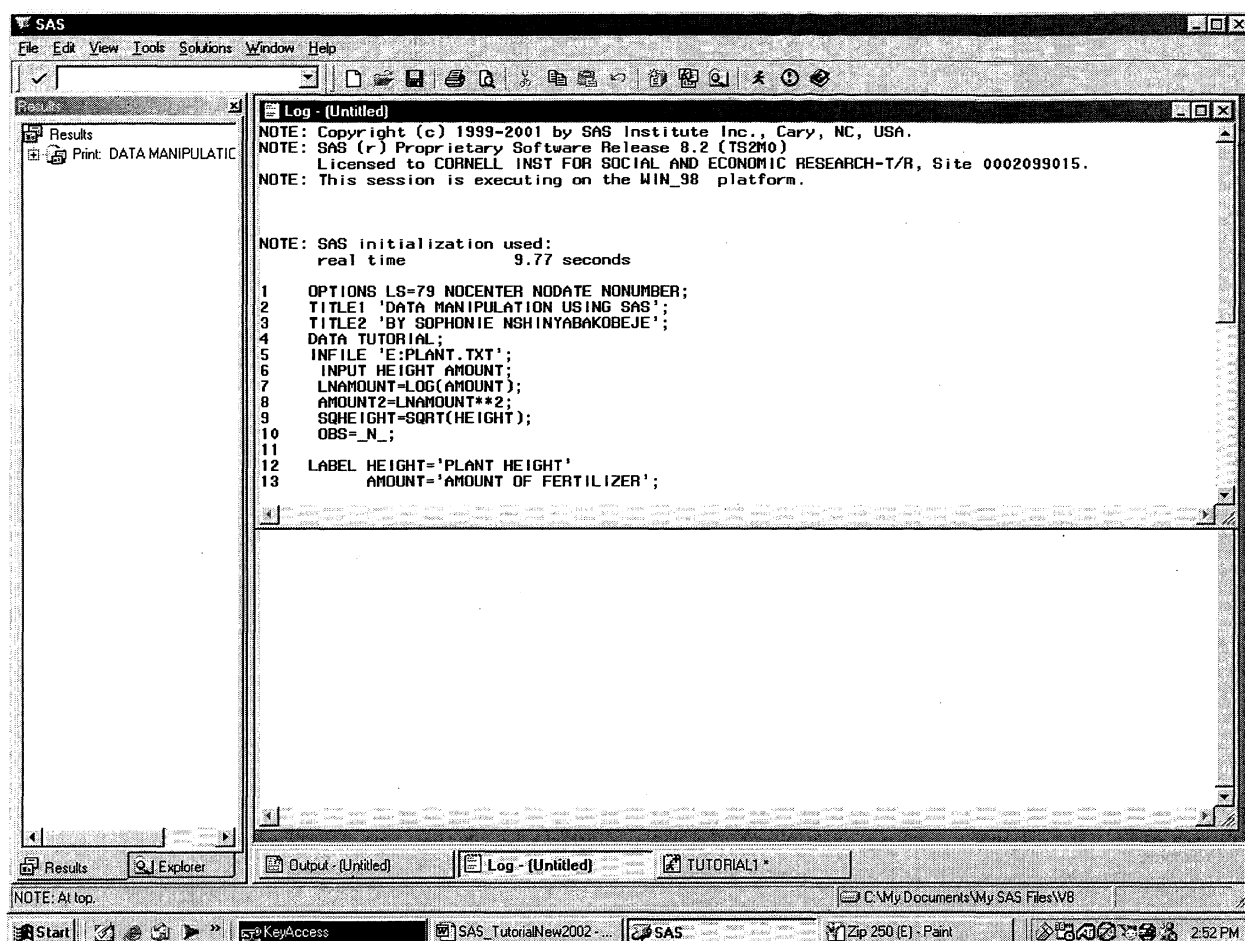
Note: To display labeling information for each variable, use the following SAS command:

```
PROC PRINT DATA=TUTORIAL LABEL;
```

The SAS output from the SAS program is given in the output window above.

- ☐ Your SAS output should look as shown in the output window above.
- ☐ Clear the Output window as follows: Choose **Edit** ➤ **Clear All**.
- ☐ Open the Log Window by choosing **Window** ➤ **Log** or by simply clicking on the Log-(Untitled) icon at the window bar. SAS displays in the Log Window results of commands processing and/or error messages, along with the time used to process SAS commands.

The log window is provided below.



It is recommended to always clear the output and the log windows since SAS continues to append information from different runs. **So please make sure to clear these windows as often as needed.**

- ❑ Clear the Log Window as follows: Choose **Window** ➤ **log** then Choose **Edit** ➤ **Clear All**.

Using SAS Function Keys

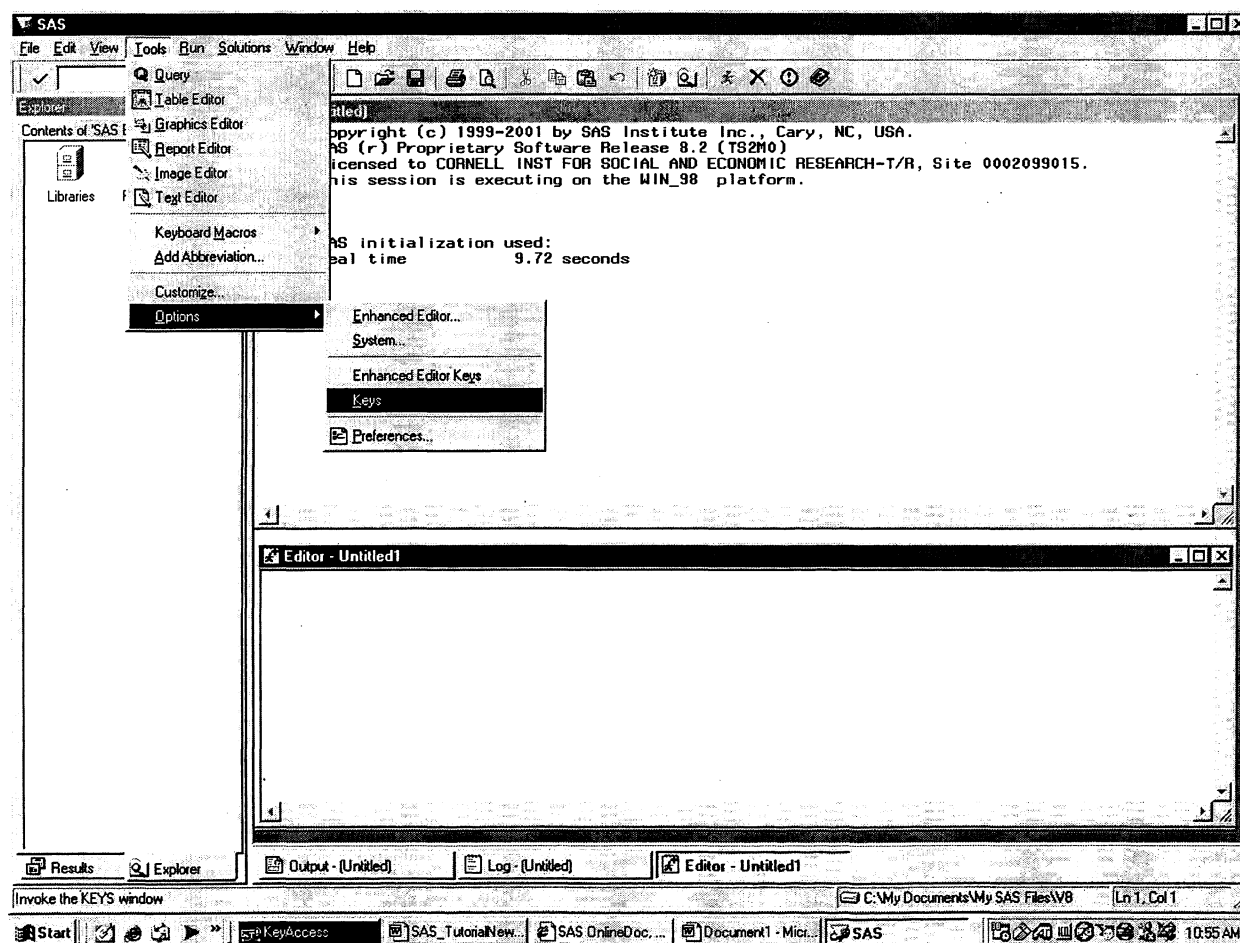
It is sometime handy to use SAS default key settings. For instance, to get help in SAS just press key **F1**. A dialog box pops up where to enter the topic of interest on which help is needed. SAS System default key settings are presented below: To submit (run) a group of SAS commands in a file, press **F3**. The following keys are used to open SAS windows: **F5** for the Program Editor Window, **F6** for the Log Window, and **F7** for Output Window. It is convenient to use SAS function when writing and running SAS programs. To see SAS function key settings, press **F9** or use the menu path: **Tools** ➤ **Options** ➤ **Keys**.

A list of SAS System's default key settings are presented in the table below.

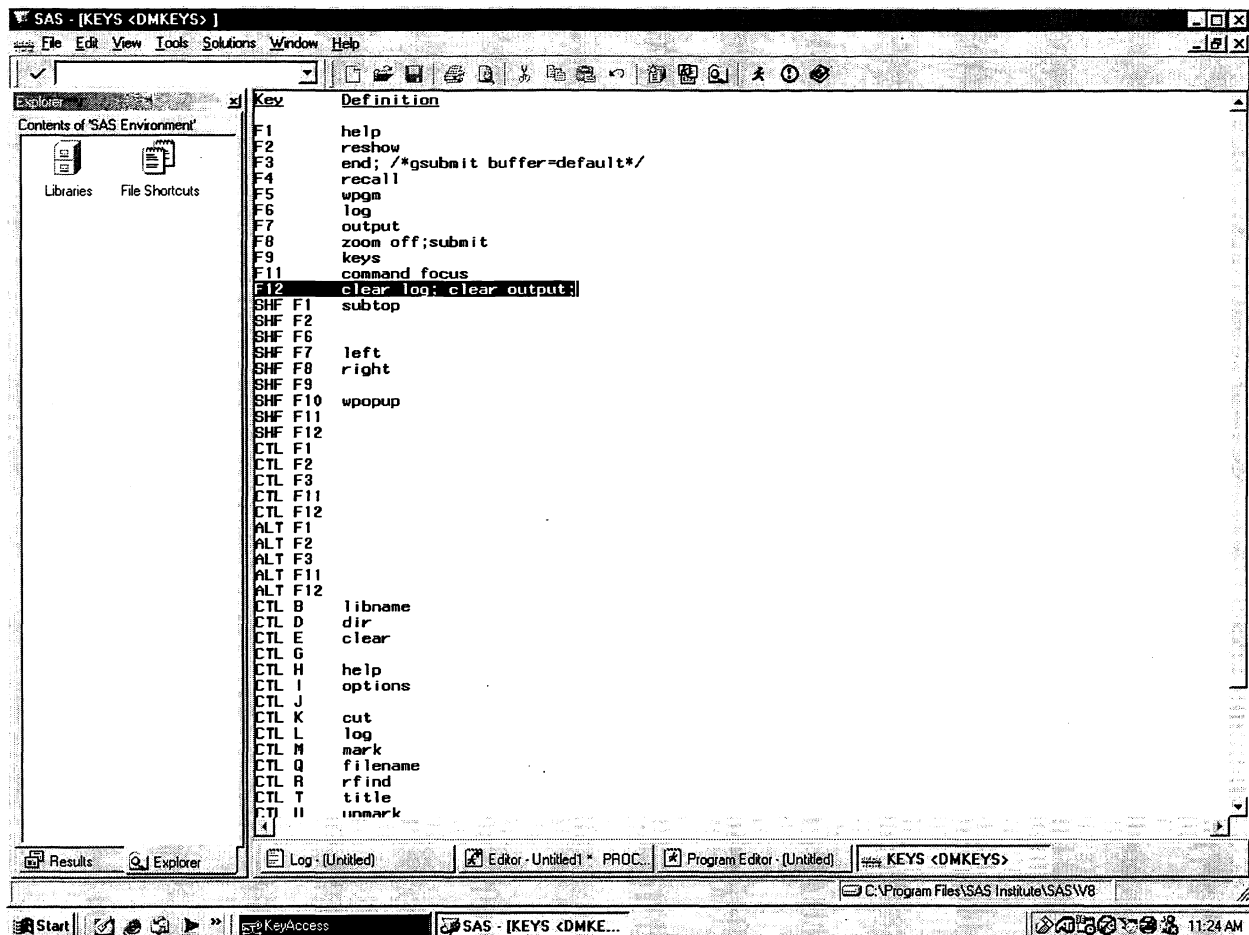
KEY	DEFAULT SETTING
F1	Help
F2	Reshow
F3	Run SAS
F4	Recall
F5	Program editor window
F6	Log window
F7	Output window
F8	Zoom off, submit
F9	Function keys
F11	Command bar
F12	

Now we need to set key F12 to clear the Editor and log windows. Proceed as follows:

❑ Tools > Options > Keys



- ❑ In the table below, go in the column next to F12 under heading definition and type **clear log; clear output;** as shown below.



- ❑ Choose: **Tools > Options > Preferences > Check Save settings on Exit > Click OK.**
- ❑ Close the SAS Keys window above by clicking on the ☐ sign at the upper right-hand side.

Creating New Variables In SAS

We need transform variables HEIGHT (square root transformation) and AMOUNT (log-transformation).

- ❑ Recall the SAS file in the Program Editor window by pressing F4 or by Choosing **Run > Recall Last Submit**
- ❑ Add the following lines below to your SAS program file, after line INPUT HEIGHT AMOUNT.

```
SQHEIGHT=SQRT(HEIGHT);
```

```
LNAMOUNT=LOG(AMOUNT);
```

```
AMOUNT2=LNAMOUNT**2;
```

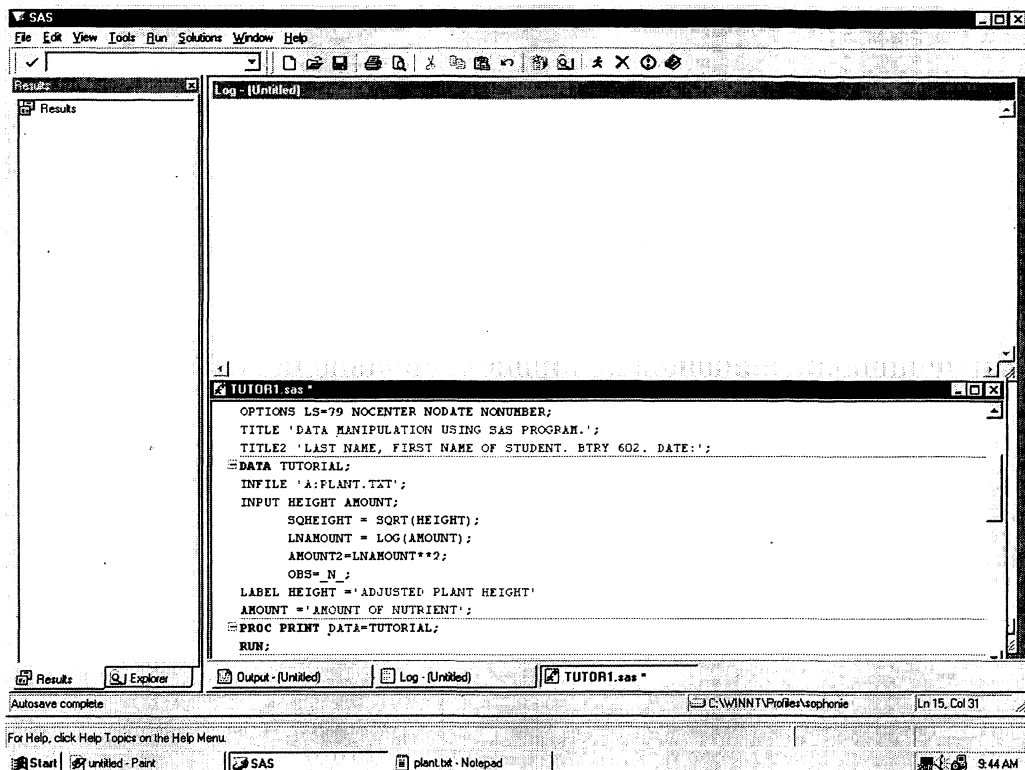
```
OBS=_N_;
```

```
RUN;
```

```
QUIT;
```

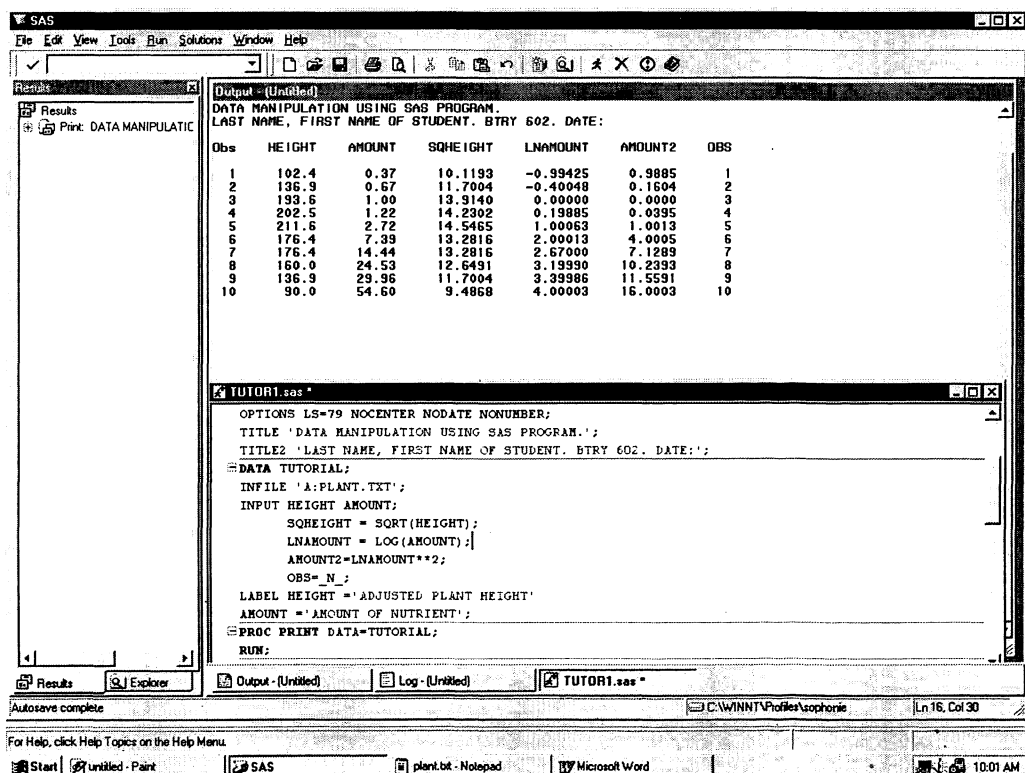
The SAS function SQRT calculates the square root of variable HEIGHT, and the LOG function calculates the NATURAL log of variable AMOUNT. Variable AMOUNT2=LNAMOUNT² and variable OBS gives each observation number for easy reference.

Your Program Editor Window should look as follows.



- Choose **Run > Submit** to process the commands in the edited SAS program file above. Or click on the run icon at the top of the window bar.

The SAS output is shown below. We now have seven columns of data including the new variables created.



- Please, select then clear the output window above: Choose **Edit > Clear All**.

Making Scatter Plots Using SAS Commands

- ☐ First open and clear the Log Window by choosing: **Window** > **Log** then choosing **Edit** > **Clear All**.
- ☐ Reopen your SAS program file, choose **Window** > **Program Editor**.
- ☐ Choose **Run** > **Recall Last Submit**.
- ☐ Edit the SAS program by inserting the following SAS commands just before the RUN statement.

*PLOTting THE DATA;

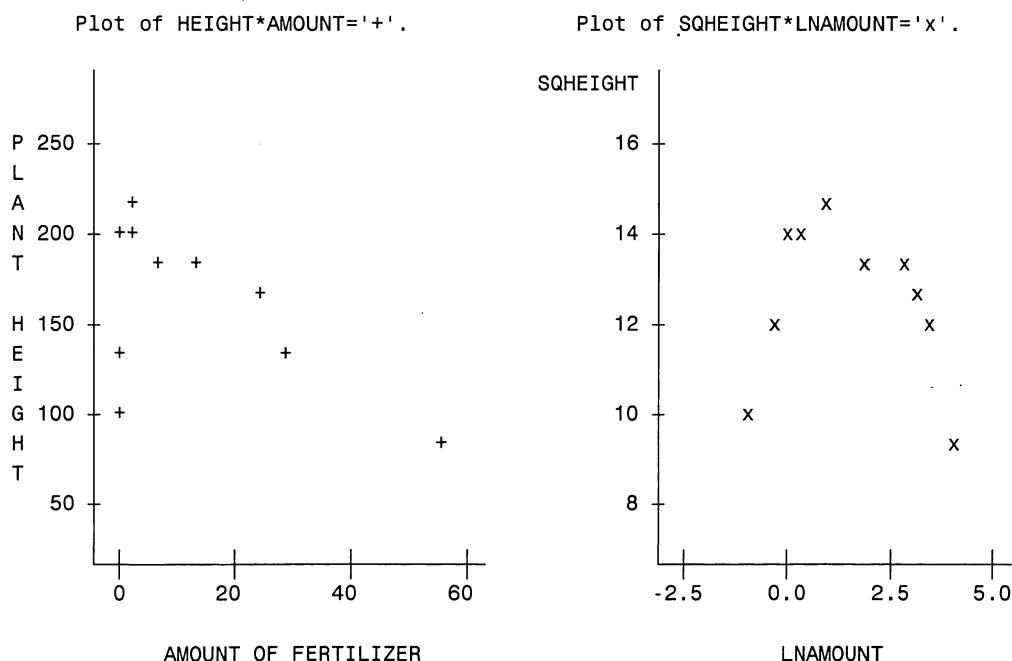
What we have above between symbols * and ; is a SAS comment.

```
PROC PLOT DATA=TUTORIAL HPERCENT=50 VPERCENT=50;  
    PLOT HEIGHT*AMOUNT='+' SQHEIGHT*LNAMOUNT='x';  
RUN;  
QUIT;
```

PROC PLOT uses the data WORK.TUTORIAL to create a scatter plot of HEIGHT against AMOUNT using the plotting symbol +. Another scatter plot is of SQHEIGHT against LNAMOUNT is created using the plotting symbol x. With Options HPERCENT=50 and VPERCENT=50, one gets plots with half of the default sizes. The width and length of the page are equal to 50% of the default sizes.

- ☐ To make the scatter plots above, highlight the three commands above and click on the run icon.

The output from SAS are given below.



- ☐ Save the SAS program file as follows: **File** > **Save As** > **A:\TUTOR1.SAS** > Select **Replace**. It is **important** to use **SAVE As** followed by **Replace**, otherwise SAS will append the newly revised file to the old file.

Checking Errors In A SAS Program File

To have an idea as to how SAS handles errors, we are going to modify the small SAS program already created. We will intentionally make a typo in our program to see how SAS reacts to it. The portion of the SAS program modified is in boldface characters (see below). In your SAS program go to the line PROC PLOT and change it to PROC **PLOTS**. Run SAS Program by choosing **Run > Submit**. Below is the SAS program file you just created.

```
/*  
THIS PROGRAM IS USED TO CREATE A SMALL SAS PROGRAM.  
TOPICS OF INTEREST INCLUDE: DATA MANIPULATION,  
PLOTING DATA USING SAS INTERACTIVE, PLOTING DATA USING SAS  
COMMANDS, AND REGRESSION ANALYSIS.  
WRITTEN BY: LAST NAME, FIRST NAME OF STUDENT.  
DATE: JANUARY 2000.  
*/  
OPTIONS LS=79 NOCENTER NODATE NONUMBER;  
TITLE 'DATA MANIPULATION USING SAS PROGRAM.';  
TITLE2 'LAST NAME, FIRST NAME OF STUDENT. BTRY 602. DATE:';  
DATA TUTORIAL;  
INFILE 'A:PLANT.TXT';  
INPUT HEIGHT AMOUNT;  
    SQHEIGHT = SQRT(HEIGHT);  
    LNAMOUNT = LOG(AMOUNT);  
    AMOUNT2=LNAMOUNT**2;  
    OBS=_N_;  
LABEL HEIGHT ='ADJUSTED PLANT HEIGHT'  
AMOUNT ='AMOUNT OF NUTRIENT';  
PROC PRINT DATA=TUTORIAL;  
*PLOTING THE DATA;  
PROC PLOTS DATA=TUTORIAL HPERCENT=50 VPERCENT=50;  
    PLOT HEIGHT*AMOUNT='+' SQHEIGHT*LNAMOUNT='x';  
RUN;  
QUIT;
```

After running SAS Program, open the Log window, to check the error message from SAS. Scroll through the content of the log window, where you read the following after Line 33. The Line number may be different depending on whether or not you continuously cleared the Log Window repeatedly. So scroll through the Log Window to locate the error message.

```
33    PROC PLOTS DATA=TUTORIAL;
```


ERROR: Procedure PLOTS not found.

SAS found an error at Line 33. This is no surprise since there is a typo in PROC PLOTS

The whole content of the log window is provided below.

```
19 DATA TUTORIAL;
20 INFILE 'A:PLANT.TXT';
21 INPUT HEIGHT AMOUNT;
22     SQHEIGHT = SQRT(HEIGHT);
23     LNAMOUNT = LOG(AMOUNT);
24     AMOUNT2=LNAMOUNT**2;
25     OBS=_N_;
26 LABEL HEIGHT ='ADJUSTED PLANT HEIGHT'
27 AMOUNT ='AMOUNT OF NUTRIENT';

NOTE: The infile 'A:PLANT.TXT' is:
      File Name=A:\PLANT.TXT,
      RECFM=V,LRECL=256

NOTE: 10 records were read from the infile 'A:PLANT.TXT'.
      The minimum record length was 19.
      The maximum record length was 20.
NOTE: The data set WORK.TUTORIAL has 10 observations and 6 variables.
NOTE: DATA statement used:
      real time          3.86 seconds
      cpu time           0.13 seconds

28 PROC PRINT DATA=TUTORIAL;
29 RUN;

NOTE: There were 10 observations read from the data set WORK.TUTORIAL.
NOTE: PROCEDURE PRINT used:
      real time          0.03 seconds
      cpu time           0.03 seconds

30 PROC PLOT DATA=TUTORIAL;
31     PLOT HEIGHT*AMOUNT='+' SQHEIGHT*LNAMOUNT='1';
32 RUN;

NOTE: There were 10 observations read from the data set WORK.TUTORIAL.
NOTE: PROCEDURE PLOT used:
      real time          26.34 seconds
      cpu time           0.05 seconds

33 PROC PLOTS DATA=TUTORIAL HPERCENT=50 VPERCENT=50;
ERROR: Procedure PLOTS not found.
34     PLOT HEIGHT*AMOUNT='+' SQHEIGHT*LNAMOUNT='1';
35 RUN;

NOTE: The SAS System stopped processing this step because of errors.
NOTE: PROCEDURE PLOTS used:
      real time          0.09 seconds
      cpu time           0.00 seconds
```

Note:

- ☐ Depending on how often you cleared the log window, you may get the message at a different line number. However, this is not of concern. The important point here is that SAS Program will locate the error in your program and give an error message.
- ☐ A handy way of running a SAS program is to highlight the SAS program's portion of interest and click on the run button.
- ☐ You can open any window by clicking on the corresponding tab at the window bar.
- ☐ You can close any window by right-clicking on the corresponding tab at the window bar then choosing close.

General Observations

- Every SAS statement ends with a semicolon ;. You may continue several statements on two or more lines. Forgetting the semicolon (;) leads to error messages which are hard to decipher. **This is always the first thing to check if your program does not run.**
- The next most common error is misspelling a SAS command, or a variable name. SAS variable names can be no more than 8 characters long.
- The third most common error is trying to use a variable that is not available. For example, this error can occur if you try to draw residual plot, but forgot to store the residuals from the regression.
- SAS ignores extra blanks, including blank lines, so you can space your program so that it is neat and easy to read.
- You can put more than one statement on a line, but this usually makes your program hard to read. SAS program files can get quite long, so it is useful to keep them readable.

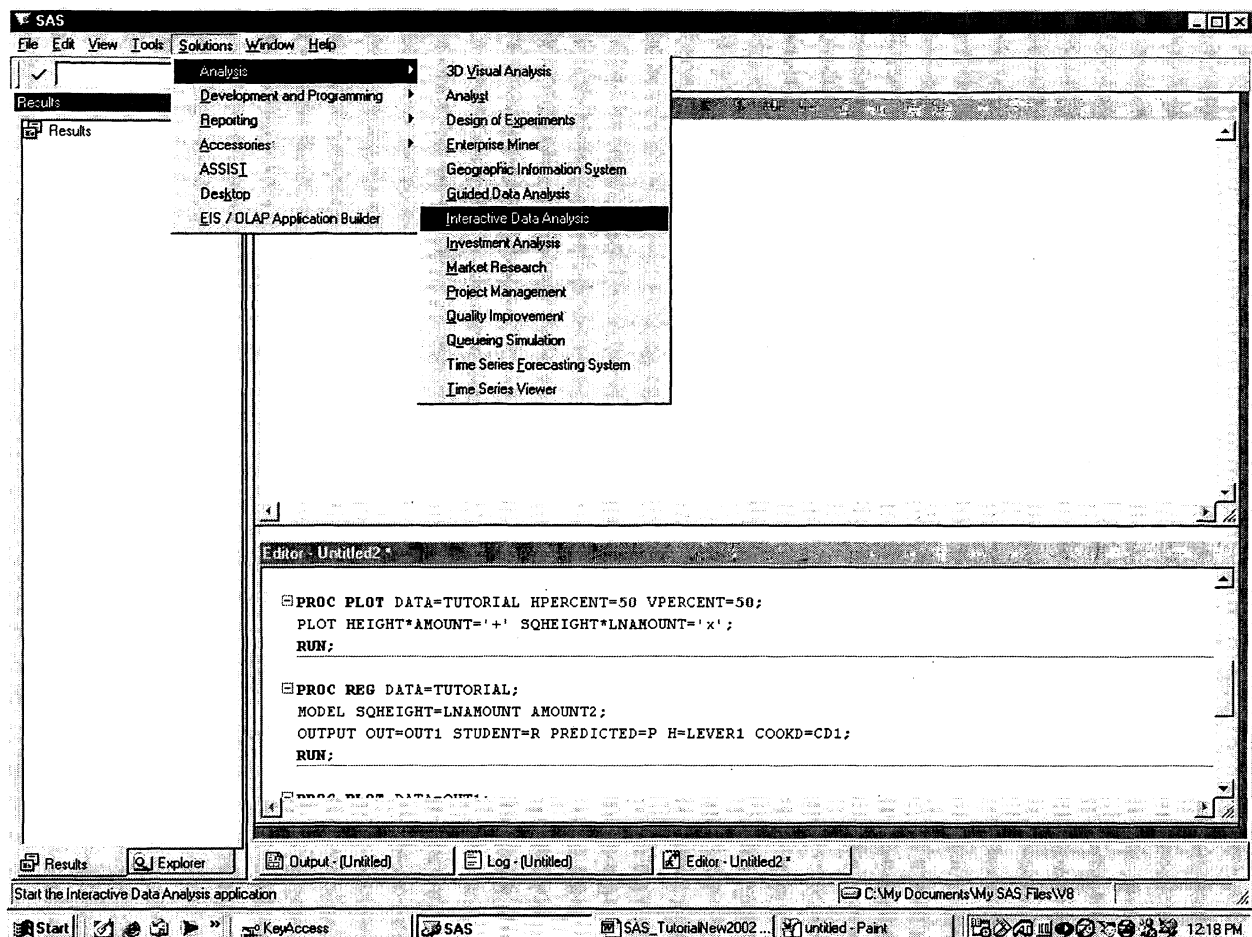
Proceed as follows:

- ☐ Open and clear the Log Window following instructions provided previously.
- ☐ Open and clear the Output Window.
- ☐ Open the Program Editor Window following instructions provided.
- ☐ Open the SAS file TUTOR1.SAS and correct the error detected by SAS by deleting the (S) in **PLOTS**.
- ☐ Save your SAS file following instructions provided previously in this regard.
- ☐ Run SAS Program.

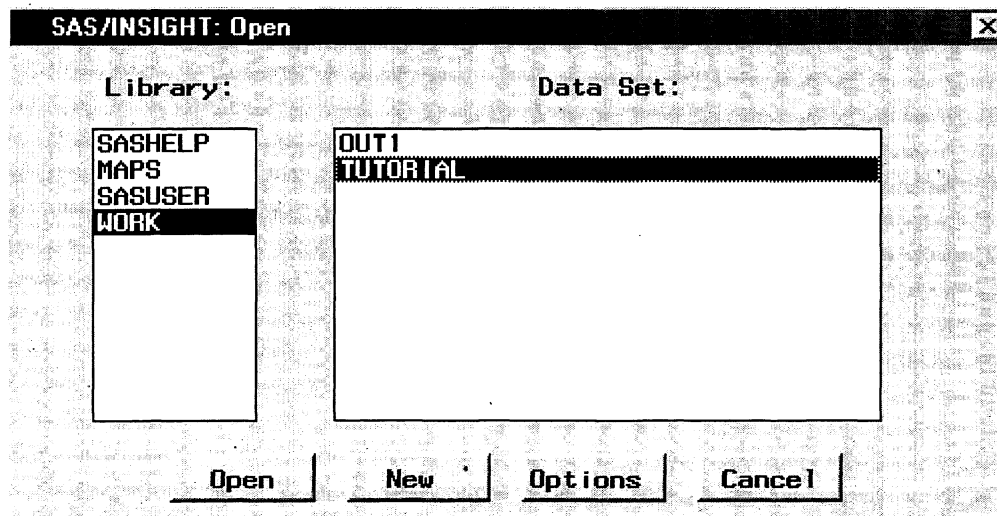
Making Scatter Plots Using SAS Interactive.

To use SAS Interactive, we need to open the temporary SAS data set called WORK.TUTORIAL. This file is stored in a Library created by SAS and called WORK. Hence the name WORK.TUTORIAL created by the statement DATA TUTORIAL and given to this temporary SAS data set. Proceed as follows to access and open WORK.TUTORIAL.

- ❑ To access SAS Interactive, proceed as follows: Choose **Solutions** ➤ **Analysis** ➤ **Interactive Data Analysis**.

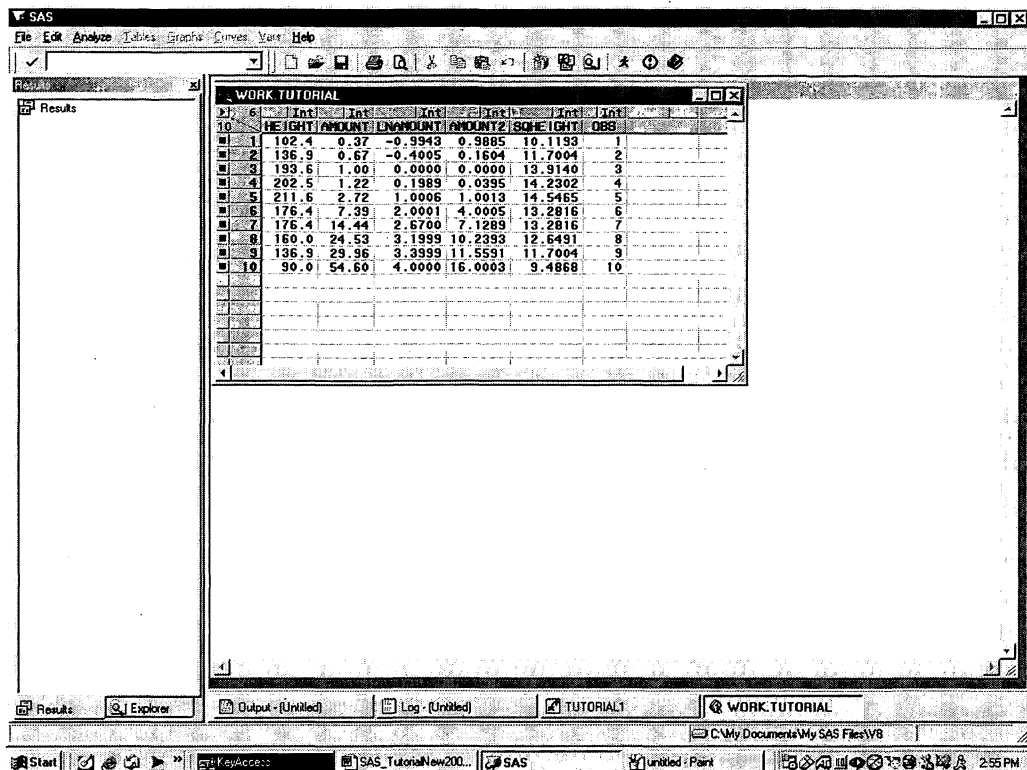


- ❑ You are prompted to the following dialog box.



- ❑ Under Library, select **Work** ➤ Under Data Set, the name **TUTORIAL** assigned to the temporary data set should appear ➤ Select the data set name **TUTORIAL** and click **Open**.

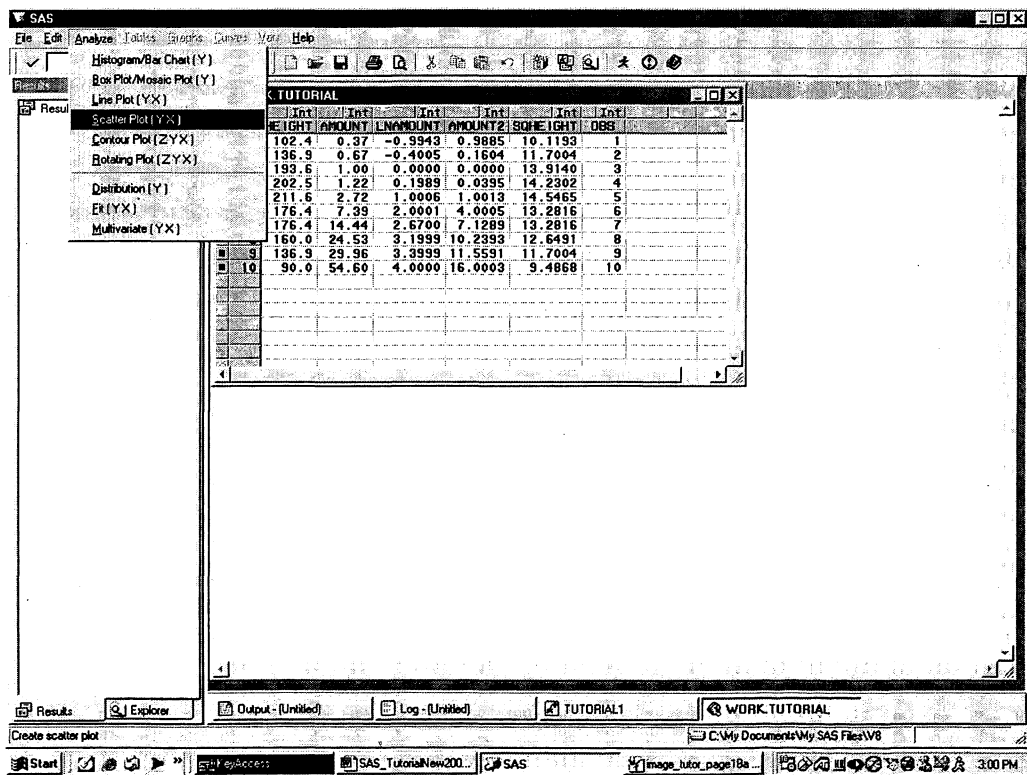
You are prompted to SAS temporary data set WORK.TUTORIAL shown on the spreadsheet below and showing all variables created with the INPUT statement.



The screenshot shows the SAS WORK.TUTORIAL data set window. The table contains 10 rows of data with the following columns: HEIGHT, AMOUNT, LNAMOUNT, AMOUNT2, SQHEIGHT, and OBS.

	HEIGHT	AMOUNT	LNAMOUNT	AMOUNT2	SQHEIGHT	OBS
1	102.4	0.37	-0.9943	0.9885	10.1193	1
2	136.9	0.67	-0.4005	0.1604	11.7004	2
3	193.6	1.00	0.0000	0.0000	13.9140	3
4	202.5	1.22	0.1989	0.0395	14.2302	4
5	211.6	2.72	1.0006	1.0013	14.5465	5
6	176.4	7.39	2.0001	4.0005	13.2816	6
7	176.4	14.44	2.6700	7.1289	13.2816	7
8	160.0	24.53	3.1999	10.2393	12.6491	8
9	136.9	29.96	3.3999	11.5591	11.7004	9
10	90.0	54.60	4.0000	16.0003	9.4868	10

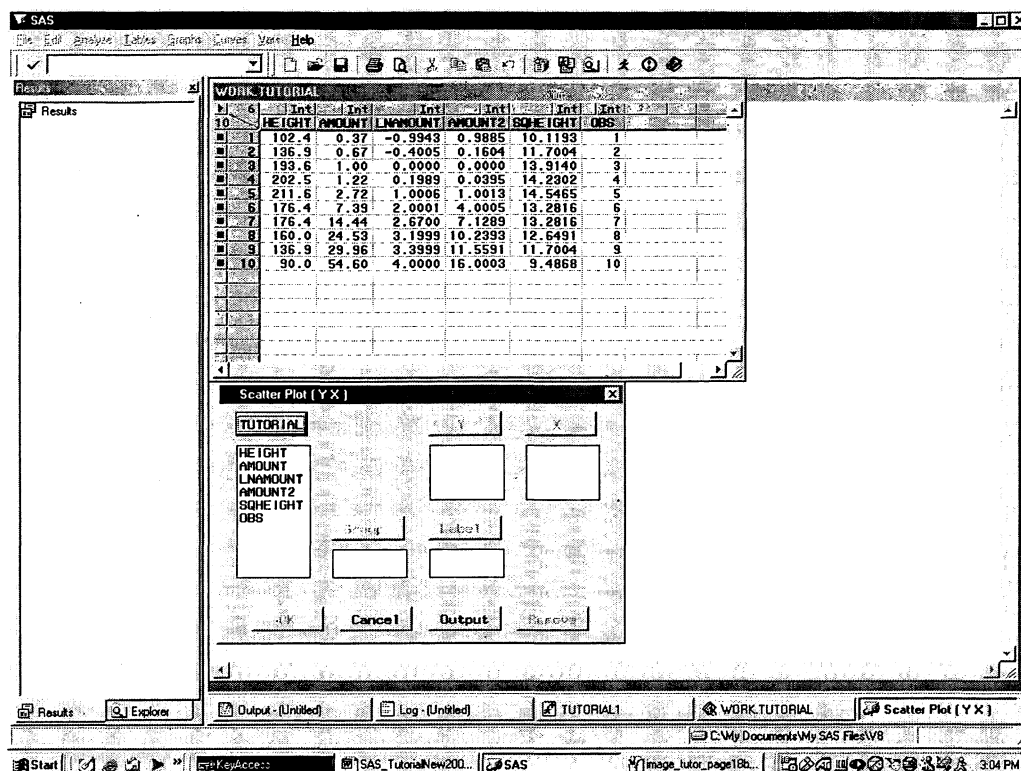
From the menu in the window above, choose **Analyze** ➤ **Scatter Plot**.



The screenshot shows the SAS WORK.TUTORIAL data set window with the **Analyze** menu open. The **Scatter Plot (YX)** option is selected. The table data is the same as in the previous screenshot.

	HEIGHT	AMOUNT	LNAMOUNT	AMOUNT2	SQHEIGHT	OBS
1	102.4	0.37	-0.9943	0.9885	10.1193	1
2	136.9	0.67	-0.4005	0.1604	11.7004	2
3	193.6	1.00	0.0000	0.0000	13.9140	3
4	202.5	1.22	0.1989	0.0395	14.2302	4
5	211.6	2.72	1.0006	1.0013	14.5465	5
6	176.4	7.39	2.0001	4.0005	13.2816	6
7	176.4	14.44	2.6700	7.1289	13.2816	7
8	160.0	24.53	3.1999	10.2393	12.6491	8
9	136.9	29.96	3.3999	11.5591	11.7004	9
10	90.0	54.60	4.0000	16.0003	9.4868	10

You are prompted to the dialog box below.

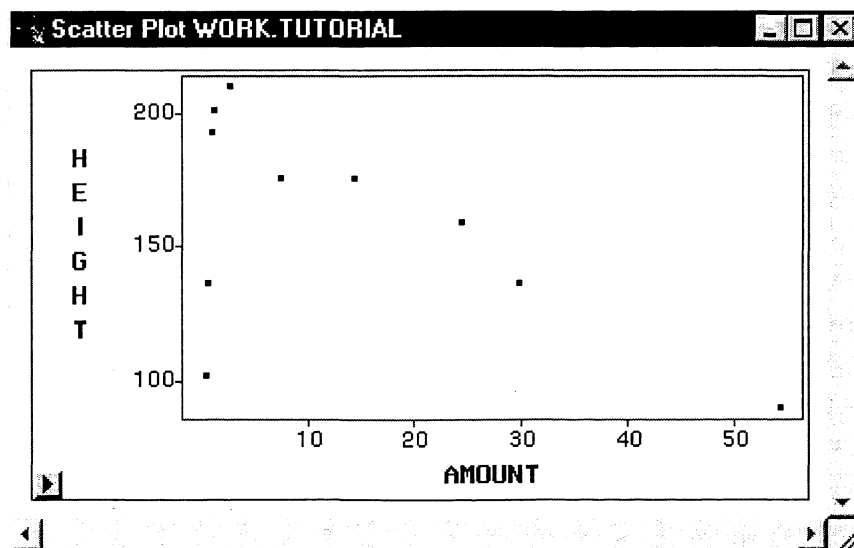


Suppose we want to make two scatter plots respectively with HEIGHT plotted against AMOUNT in the first plot and SQHEIGHT plotted against LNAMOUNT in the second plot.

- ☐ To plot HEIGHT against AMOUNT, proceed as follows:
- ☐ Select HEIGHT, then click Y
- ☐ Select AMOUNT then click X, then click OK.

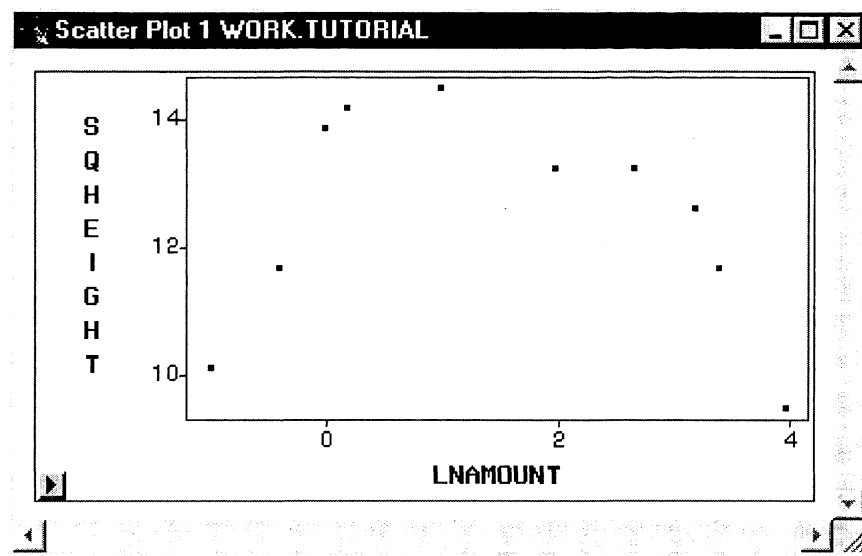
You will obtain a scatter plot HEIGHT against AMOUNT as shown below.

Scatter plot of HEIGHT against AMOUNT.



- ☐ To plot SQHEIGHT against LNAMOUNT, proceed as follows:
- ☐ Select SQHEIGHT, then click Y
- ☐ Select LNAMOUNT then click X, then click OK. The scatter plot obtained is given below.

Scatter plot of SQHEIGHT against LNAMOUNT.



Note that the transformation of the two variables enables to have a more tangible non-linear relationship between SQHEIGHT and LNAMOUNT.

- ☐ Close each graph by clicking on it and checking the sign ☒ at the upper right-hand side corner of the box around the graph.

Making A Scatter Plot With Loess Smooth Using SAS Interactive

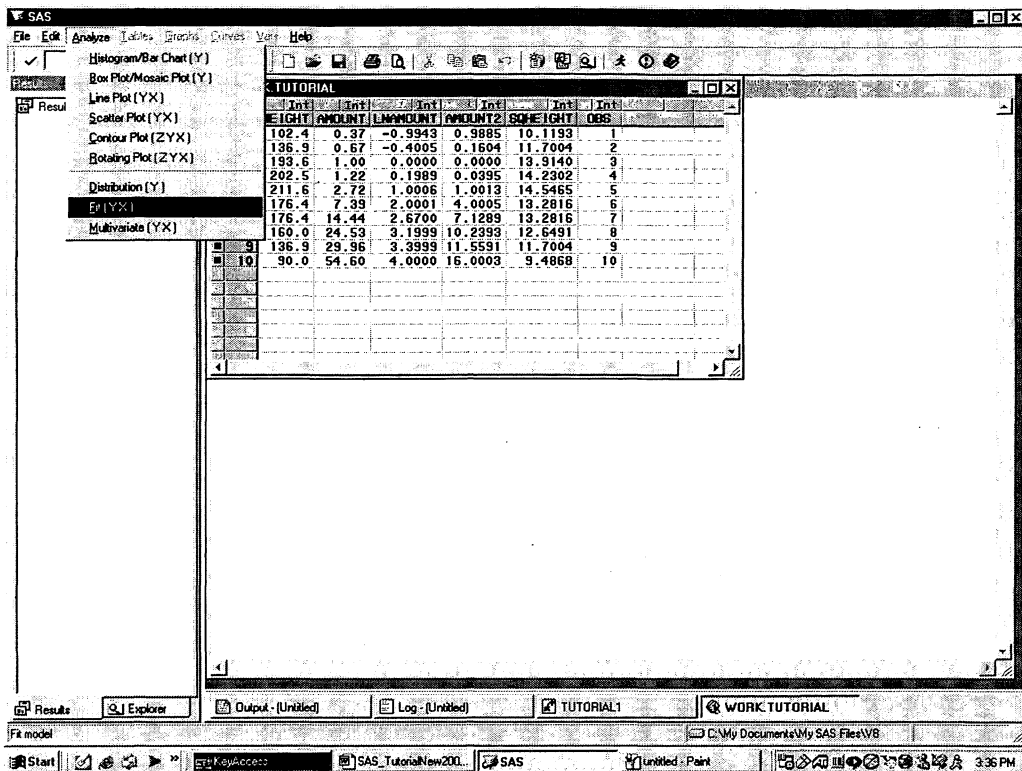
Scatter plots with loess smooth may help get better insights into the relationship between two variables. Both the parametric and non-parametric smoothing method will be illustrated here. First, the parametric method is illustrated. Variables SQHEIGHT and LNAMOUNT will be used to make a scatter plot with loess smooth.

Parametric Method

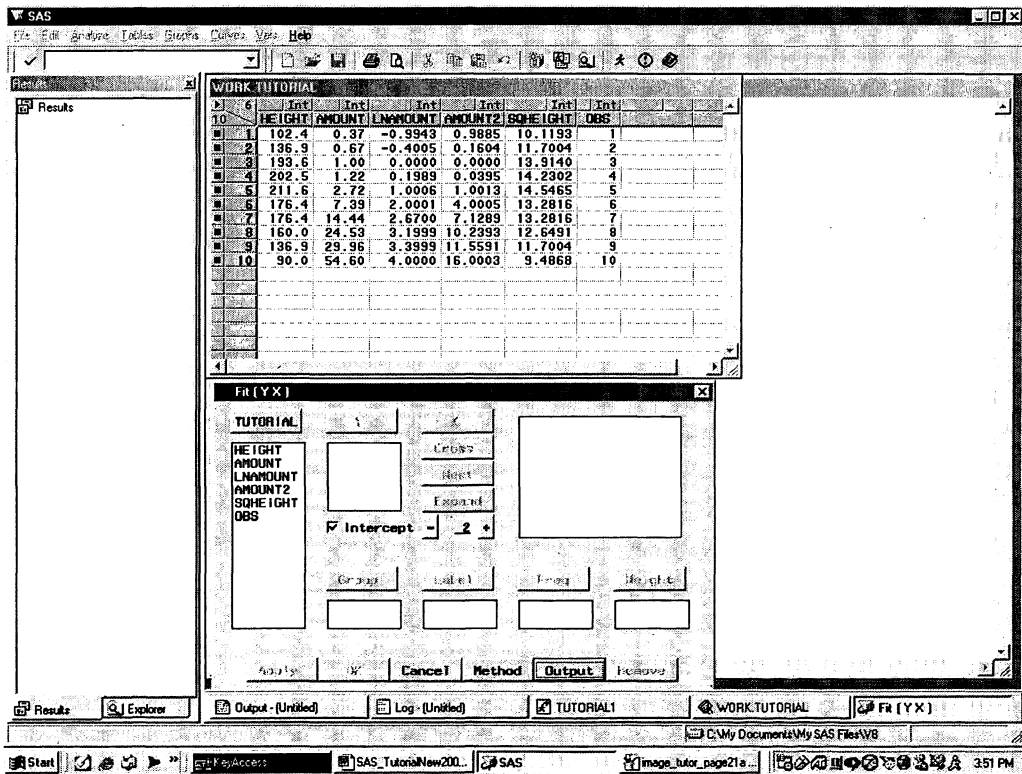
Quadratic smoothing will be performed because the scatter plot above shows a quadratic relationship between variables SQHEIGHT and LNAMOUNT. The data set WORK.TUTORIAL is still open.

To make a parametric loess smooth graph, proceed as follows:

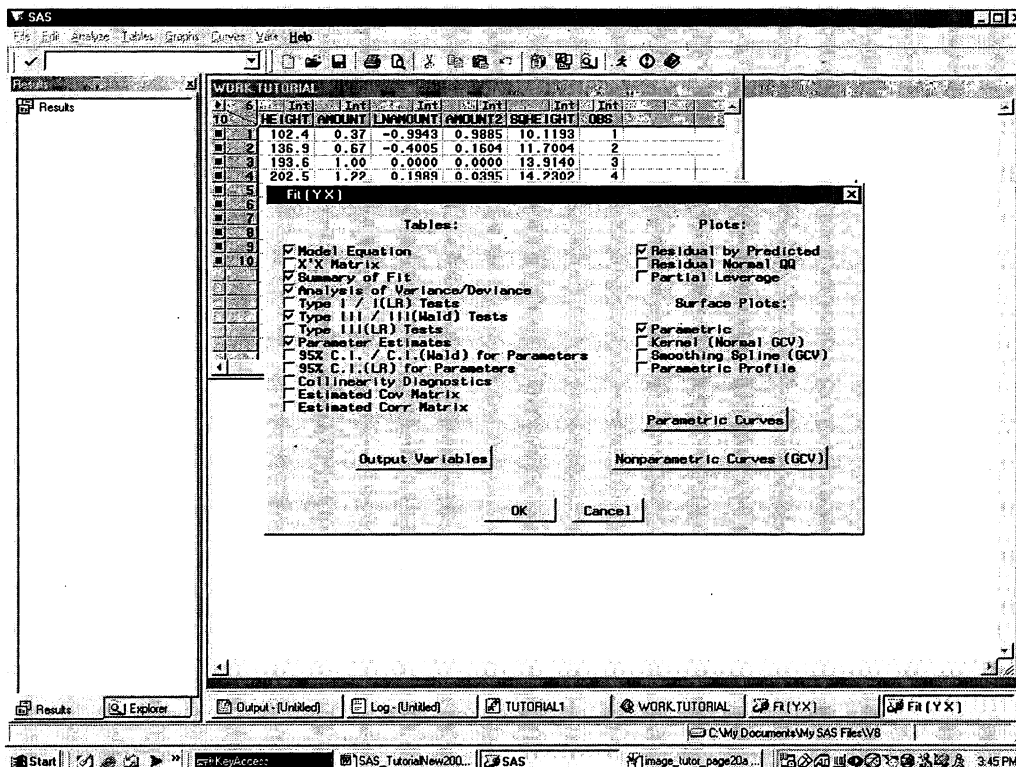
- ❑ Choose **Analyze** ➤ **Fit(Y X)**.



You are prompted to the following dialog box.

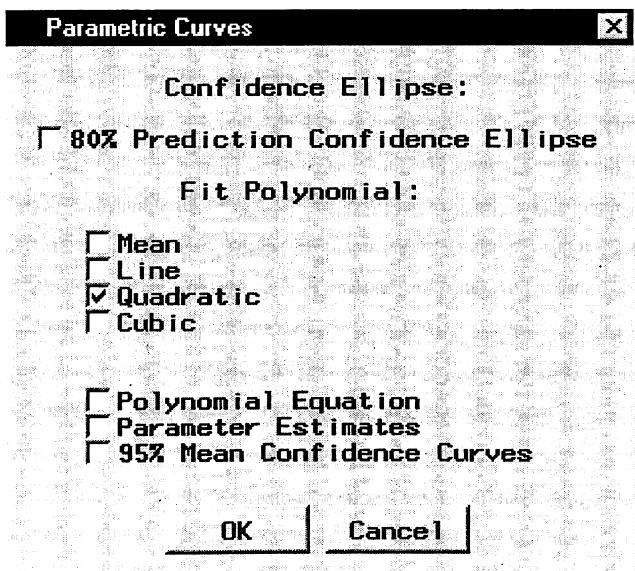


Click on the Output tab and unselect all checked boxes below.



By unselecting the boxes above, you limit the amount of output from SAS Interactive.

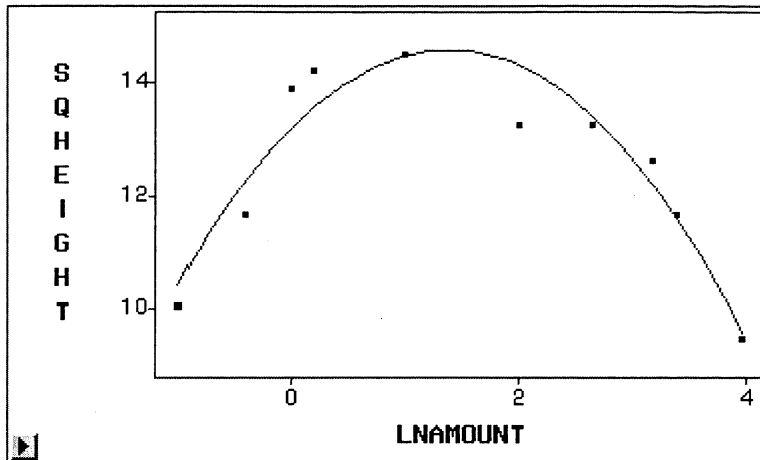
- ☐ Click on the Parametric Curves tab in the dialog box above.
- ☐ Deselect Line in the dialog below because we do not want to fit a line to the data. Check Quadratic to fit a quadratic model to the data between variables SQHEIGHT and LNAMOUNT as justified by the scatter plot between these two variables (see Page 20).



- ☐ Click OK twice to open a dialog box used for selecting plotting variables.
- ☐ You are prompted to the dialog box below and you need to select the two plotting variables.

- ☐ Select SQHEIGHT, then click on Y.
- ☐ Select LNAMOUNT, then click on X.
- ☐ Click on the Apply tab to generate the plot.

The parametric loess smooth graph obtained is given below.

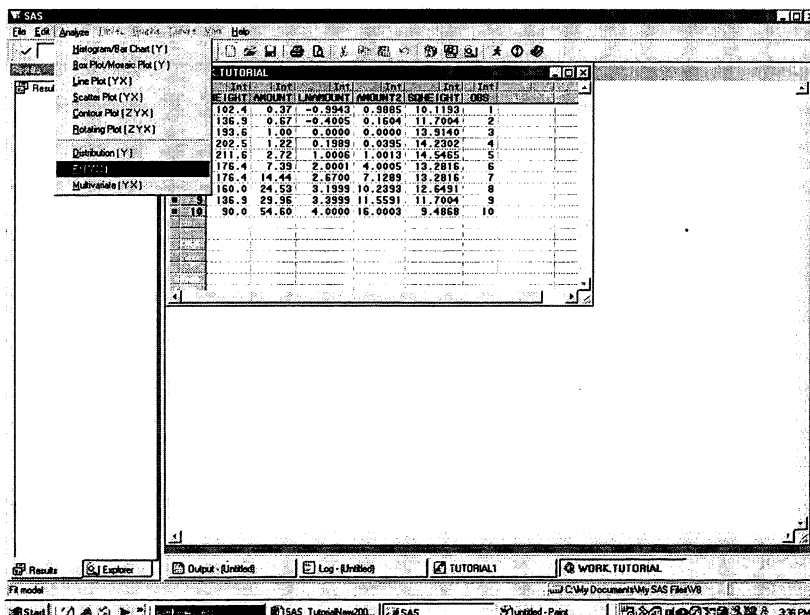


- ☐ Close the non-parametric loess smooth graph above by clicking on **x** at the top of the upper right-hand side of the graph's box.

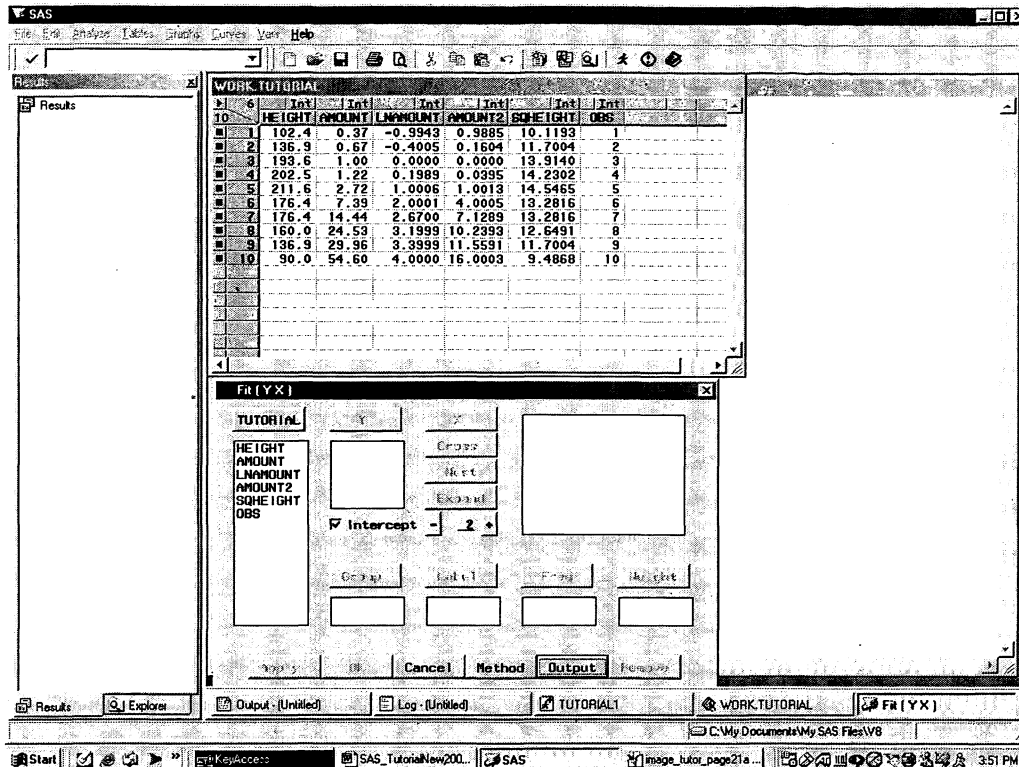
Non-parametric Method

To make a non-parametric loess smooth, the steps are similar to those described above, to some extent. We will make a non-parametric loess smooth on a scatter plot of SQHEIGHT against LNAMOUNT.

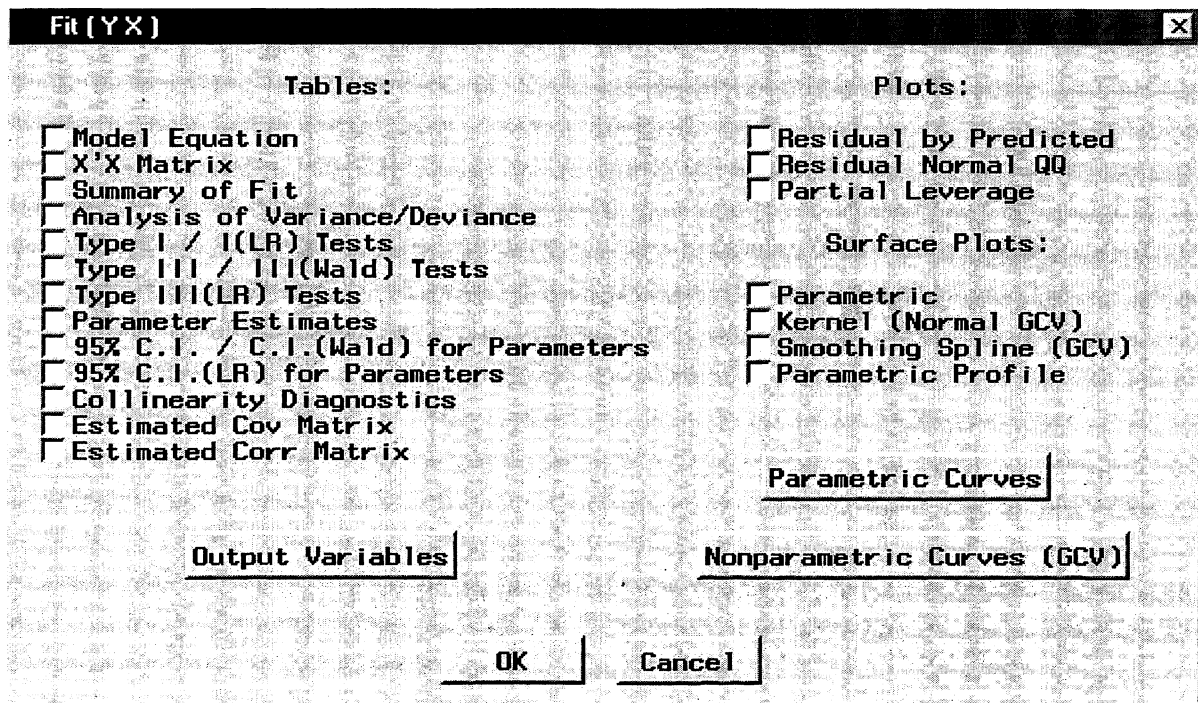
- ☐ Choose Analyze ► Fit(Y X).



You are prompted to the following dialog box.



- ❑ Deselect all checked boxes to reduce the amount of SAS Interactive output as shown below.



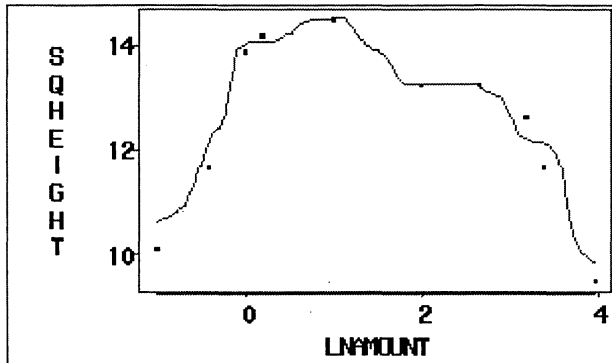
- ❑ To disable the parametric option, click on the parametric tab, then deselect checked boxes if any. Otherwise, a parametric curve will appear along with the non-parametric curve on the graph.


- ☐ Click on the Nonparametric Curves tab, then under local polynomial, check **method loess** , **type mean** and **weight tri-cube**. Please, deselect Linear because the relationship between the two variables is curved (nonlinear).
- ☐ Click OK twice.

You are prompted to select the two plotting variables as you did in the parametric case above.

- ☐ Select SQHEIGHT as the response variable, then click on Y.
- ☐ Select LNAMOUNT as the predictor variable, then click on X.
- ☐ Click on the Apply tab to generate the plot.

The plot generated using the non-parametric loess smoothing method is shown below.



- ☐ To change the smoothness of the curve above, proceed as follows:
- ☐ Scroll down and go to the bottom of the SAS Interactive output window ➤ Under **Loess Fit** change the value of **Alpha** by clicking its box arrows to the left or right until you get a smoother curve.
- ☐ Close each of the SAS Interactive windows by selecting it and clicking on  at the top upper right-hand side of each window.
- ☐ Select and clear the output window
- ☐ Select and clear the log-window
- ☐ Open TUTORIAL1.SAS file by clicking on it at the bottom of the window bar.

The next step is to perform regression analysis in SAS. You therefore need to exit SAS Interactive.

Performing Regression Analysis Using SAS

The SAS program file TUTOR1.SAS that you have created so far has the following commands:

```
/*
THIS PROGRAM IS USED TO CREATE A SMALL SAS PROGRAM.
TOPICS OF INTEREST INCLUDE: DATA MANIPULATION,
PLOTING DATA USING SAS INTERACTIVE, PLOTING DATA USING SAS
COMMANDS, AND REGRESSION ANALYSIS.
WRITTEN BY: LAST NAME, FIRST NAME OF STUDENT.
DATE: JANUARY 2000.
*/
OPTIONS LS=79 NOCENTER NODATE NONUMBER;
TITLE 'DATA MANIPULATION USING SAS PROGRAM.';
TITLE2 'LAST NAME, FIRST NAME OF STUDENT. BTRY 602. DATE: ';
DATA TUTORIAL;
```

```
INFILE 'A:PLANT.TXT';
INPUT HEIGHT AMOUNT;
      SQHEIGHT = SQRT(HEIGHT);
```

```
      LNAMOUNT = LOG(AMOUNT);
      AMOUNT2 = LNAMOUNT**2;
      OBS = _N_;
LABEL HEIGHT = 'ADJUSTED PLANT HEIGHT'
      AMOUNT = 'AMOUNT OF NUTRIENT';
PROC PRINT DATA=TUTORIAL;
*PLOTting THE DATA;
PROC PLOT DATA=TUTORIAL HPERCENT=50 VPERCENT=50;
PLOT HEIGHT*AMOUNT='+' SQHEIGHT*LNAMOUNT='X';
RUN;
QUIT;
```

The small SAS program above is edited to incorporate SAS commands for polynomial regression analysis. The scatter plot SQHEIGHT against LNAMOUNT showed that the relationship between these two variables was curvilinear. We therefore need to fit a quadratic polynomial regression model of the form

$$SQHEIGHT = \beta_0 + \beta_1 LNAMOUNT + \beta_2 LNAMOUNT^2 + \varepsilon \text{ to the data.}$$

Above the RUN statement, add the following commands required for regression analysis:

```
*PERFORMING REGRESSION ANALYSIS;
PROC REG DATA=TUTORIAL;
MODEL SQHEIGHT=LNAMOUNT AMOUNT2;
OUTPUT OUT=OUT1 STUDENT=R PREDICTED=P H=LEVER1 COOKD=CD1;
RUN;
```

- ☐ Using the mouse, highlight the five SAS lines above and click on the run icon.

PROC REG performs regression analysis using data set WORK.TUTORIAL. The MODEL statement gives the response and the predictor variables. The response variable is at the left hand side of the equal sign (=) whereas the independent variables are listed at the right hand side of the equal sign (=).

Statement OUTPUT OUT=OUT1 creates a new temporary SAS data set called WORK.OUT1 containing all initial variables in data set WORK.TUTORIAL along with the following variables from regression analysis:

- ☐ studentized residuals (STUDENT) stored in a variable named R,
- ☐ predicted values (PREDICTED) stored in a variable named P,
- ☐ leverage values (H) stored in a variable named LEVER1.
- ☐ and Cook's distances (COOKD) stored in a variable named CD1.

Note: The names of the variables at the left hand side of the equal sign "=" are created by SAS. For instance, H is a variable created by SAS to store leverage values whereas COOKD is a variable created by SAS to store Cook's distance values. The name of a SAS variable at the right hand side of the equal sign "=" as in H=LEVER1 indicate the name we assign to of a variable stored in a temporary SAS data set for a specific SAS variable of interest.

Part of the SAS output for the regression analysis given below.

Dependent Variable: SQHEIGHT

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	23.74285	11.87142	30.70	0.0003
Error	7	2.70715	0.38674		
Corrected Total	9	26.45000			

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	13.18288	0.26907	48.99	<.0001
LNAMOUNT	1	2.04995	0.31492	6.51	0.0003
AMOUNT2	1	-0.73988	0.09656	-7.66	0.0001

Now we can use PROC plot or SAS Interactive Analysis to assess validity of regression model assumptions based on diagnostic plots involving variables R, P, LEVER1 and CD1.

☐ Add the following SAS commands to your SAS program, above the RUN statement.

```
*DIAGNOSTIC PLOTS IN REGRESSION ANALYSIS;
PROC PLOT DATA=OUT1 HPERCENT=40 VPERCENT=40;
PLOT R*P = '+'/VREF=0;
PLOT LEVER1*OBS='+' CD1*OBS='+';
RUN;
```

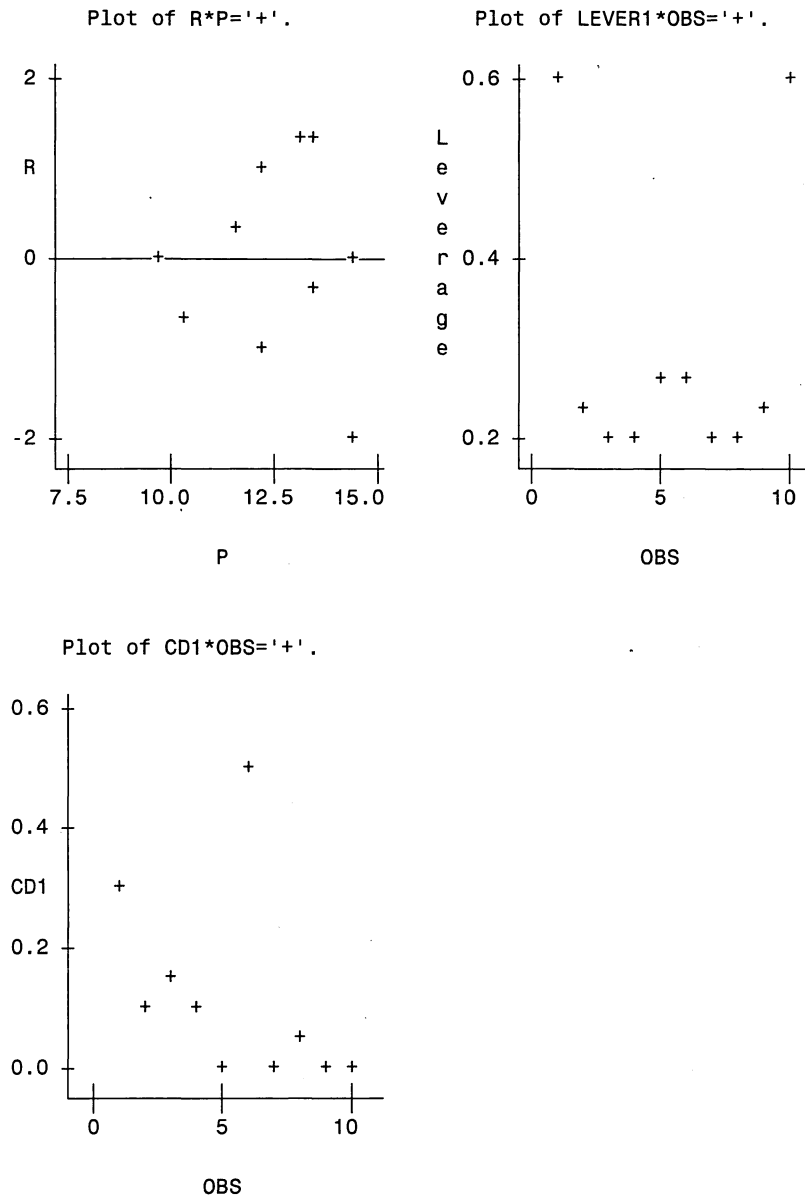
Diagnostic plots are constructed using variables stored in the temporary SAS data set WORK.OUT1. R is plotted against P.

VREF draws a reference horizontal line at the zero mean level, i.e. R=0. LEVER1 and CD1 are each plotted against OBS.

Options HPERCENT=40 and VPERCENT=40 format the plots to respectively 40% of their lengths and widths. So the page size is reduced to respectively 40% of its length and width.

☐ Using the mouse, highlight the five SAS lines above and click on the run icon.

The SAS output is shown below.



❑ Now select and clear the log-window and the output window.

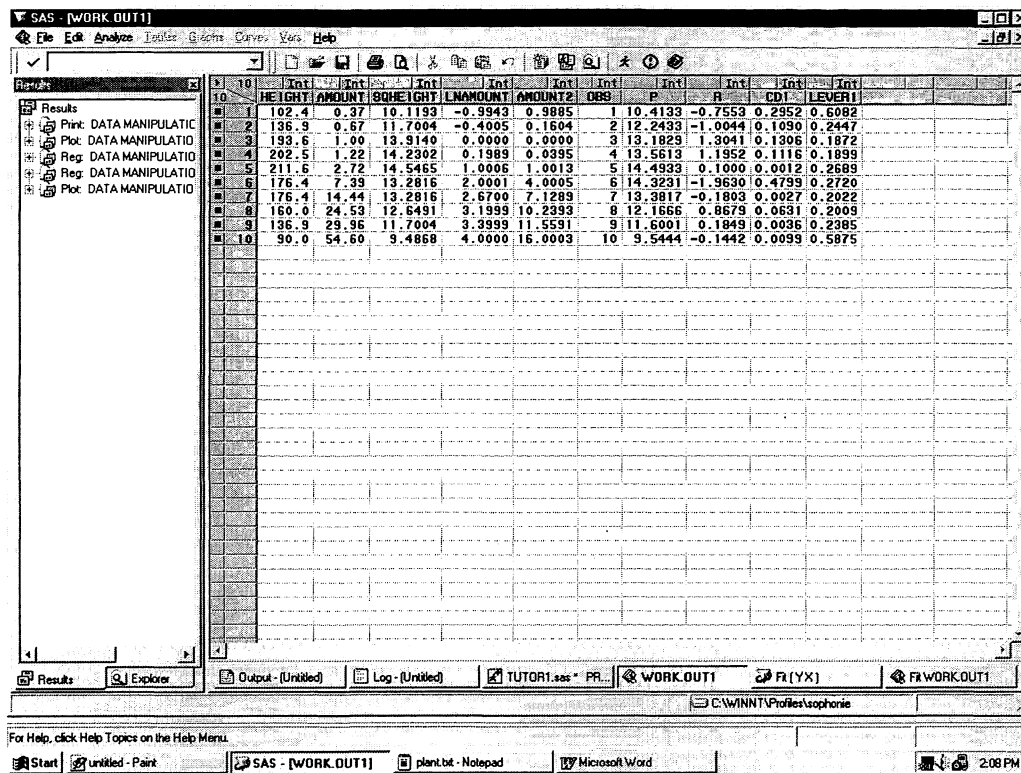
Creating Regression Diagnostic Plots Using SAS Interactive

Graphs created with SAS Interactive Analysis are better than those created using PROC PLOT. Now, you are going to create diagnostic plots using SAS Interactive Analysis. Steps needed in this regard have been discussed with variable SQHEIGHT plotted against LNAMOUNT and are summarized below (please refer to previous notes for further details if needed).

Suppose we want to make: a plot of residuals (R) against P with a nonparametric loess smooth, and a plot of Cook's distance (CD1) against observation number (OBS). These variables were stored in a SAS temporary data set WORK.OUT1 using the OUTPUT statement.

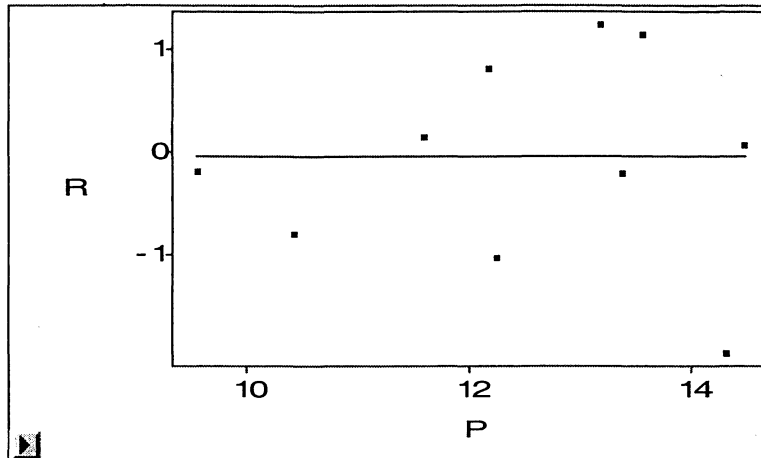
Plot Of Residuals Versus Predicted Values

- ☐ Choose Solutions ► Analysis ► Interactive Data Analysis
- ☐ Click on WORK under Library ► click on OUT1 under data set ► click on the Open tab



- ☐ Analyze ► Fit(Y X).
- ☐ Click on Output and deselect all checked boxes in the dialog box you are prompted to.
- ☐ Click on the parametric tab.
- ☐ Deselect Line in the next dialog box since we are not doing linear parametric smoothing ► Click OK.
- ☐ Click on the nonparametric tab ► check loess to perform nonparametric smoothing.
- ☐ Under Type, check Mean ► Deselect Linear ► Click OK twice.
- ☐ Under OUT1, select R, then click on Y ► Select P, then click on X.
- ☐ Click on the Apply tab to generate the plot of residuals against predicted values including a loess smooth.

The plot of residuals versus predicted values along with a nonparametric smooth is given below.



Plot Of Cook's Distance Versus Observation Numbers

- ☐ Choose Solutions ► Analysis ► Interactive Data Analysis
- ☐ Click on WORK under Library ► click on OUT1 under data set ► click on the Open tab. You will get the following spreadsheet for data SAS data set WORK.OUT1.

SAS - [WORK.OUT1]

File Edit Analyze Tables Graphics Window Help

Results

Print: DATA MANIPULATIO
Plot: DATA MANIPULATIO
Reg: DATA MANIPULATIO
Reg: DATA MANIPULATIO
Plot: DATA MANIPULATIO

	HEIGHT	AMOUNT	HEIGHT	AMOUNT	AMOUNT2	OBS	P	R	CDI	LEVER1
1	102.4	0.37	10.1193	-0.9943	0.9885	1	10.4133	-0.7553	0.2952	0.6082
2	136.9	0.67	11.7004	-0.4005	0.1604	2	12.2433	-1.0044	0.1030	0.2447
3	193.6	1.00	13.9140	0.0000	0.0000	3	13.1829	1.3041	0.1308	0.1872
4	202.5	1.22	14.2302	0.1389	0.0395	4	13.5613	1.1952	0.1116	0.1899
5	211.6	2.72	14.5465	1.0006	1.0013	5	14.4933	0.1000	0.0012	0.2689
6	176.4	7.39	13.2816	2.0001	4.0005	6	14.3231	-1.9630	0.4799	0.2720
7	176.4	14.44	13.2816	2.6700	7.1289	7	13.3817	-0.1803	0.0027	0.2022
8	160.0	24.53	12.6491	3.1999	10.2393	8	12.1666	0.8679	0.0631	0.2009
9	136.9	29.96	11.7004	3.3999	11.5591	9	11.6001	0.1849	0.0036	0.2385
10	90.0	54.60	9.4868	4.0000	16.0003	10	9.5444	-0.1442	0.0095	0.5875

Results Explorer Output - (Untitled) Log - (Untitled) TUTOR1.sas - PR... WORK.OUT1 Fx (YX) Fx WORK.OUT1

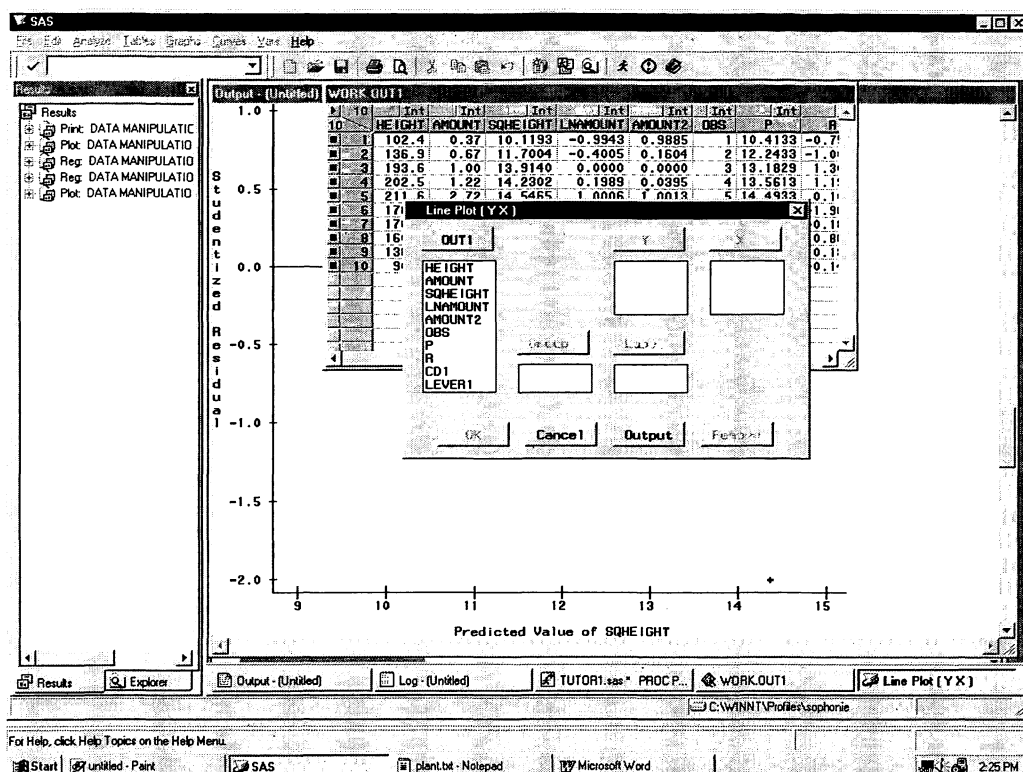
C:\WINNT\Profile\sophom...

For Help, click Help Topics on the Help Menu.

Start [SAS - [WORK.OUT1]] plant bit - Notepad Microsoft Word 2:08 PM

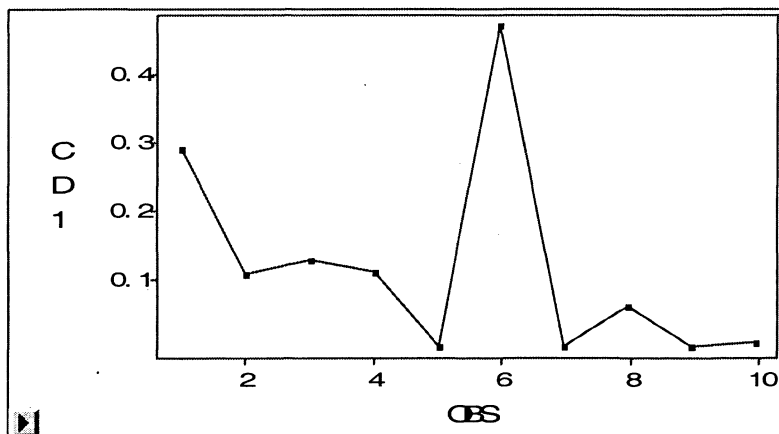
- ☐ Analyze ► Line Plot (Y X).

The following dialog box is displayed.





- ☐ Under OUT1, select CD1, then click on Y ➤ Select OBS, then click on X ➤ Click OK.
- ☐ Right-click of the plot and choose **Observations** to show data points on the graph.

The line plot of Cook's distance (CD1) against observation numbers (OBS) is shown below.



Note: A plot of leverage (LEVER1) versus observation numbers (OBS) is obtained in a similar way.

- ☐ Close each of the graph windows from SAS Interactive by clicking on the  sign at the upper right-hand side of each graph window.
- ☐ Close the SAS data set WORK.OUT1 by clicking on its window and then by clicking on the  sign at the upper right-hand side.

Assessing The Normality Assumption

- ❑ Open your SAS program file and add the following commands, just before the Run statement

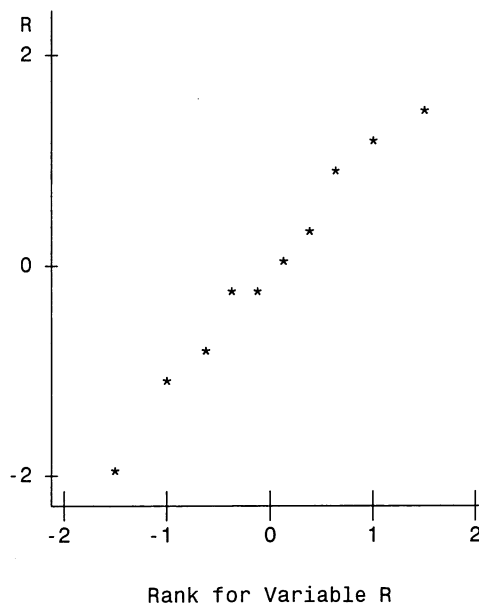
```
*NORMAL PROBABILITY PLOT;  
PROC RANK DATA=OUT1 OUT=OUT2 NORMAL=BLOM;  
VAR R;  
RANKS NR3;  
PROC PLOT DATA=OUT2 HPERCENT=50 VPERCENT=50;  
PLOT R*NR3='*';  
RUN;  
QUIT;
```

PROC RANK is used to calculate normal scores. Statement **DATA=OUT1** tells SAS to use data **WORK.OUT1**, and **OUT=OUT2** instructs SAS to store normal scores in **WORK.OUT2**. **NORMAL=BLOM** tells **PROC RANK** to compute the normal scores using Blom's method. **VAR R** statement indicates that normal scores are computed for variable **R** containing residuals. **RANKS NR3** stores normal scores in a variable named **NR3**. **PROC PLOT** makes a normal probability plot of residuals (**R**) against normal scores (**NR3**). You can make a normal probability plot with a loess smooth by opening **WORK.OUT2** (follow instructions provided in **SAS Interactive for Data Analysis**).

- ❑ Using the mouse, highlight the SAS commands above and click on the run icon.

The normal probability plot of residuals is given below.

Plot of R*NR3. Symbol used is '*'.



- ❑ Open and Save the SAS program file (use **SAVE AS**, then **REPLACE**) as explained previously.

Problem: using temporary SAS data set **WORK.OUT2** created above, use **SAS Interactive** to make a normal probability of residuals using nonparametric loess smooth with option mean.

Assessing The Equal Variance Assumption

The equal variance assumption is assessed by making a plot of absolute values of the studentized residuals against the predicted values of the dependent variable. A plot of residuals versus predicted values can also be used. Here, we will use absolute values of studentized residuals. So we need to create a temporary SAS data set which contains among other things the absolute values of residuals.

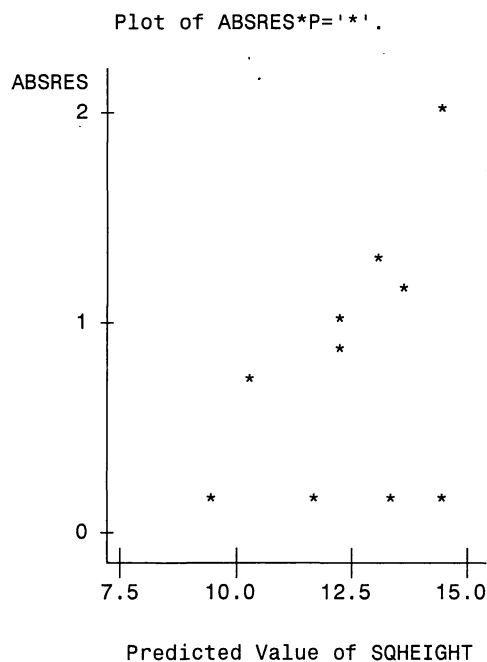
Add the following lines to your program above the RUN statement and save your file.

```
DATA OUT4;  
SET OUT1;  
ABSRES=ABS(R);  
PROC PLOT DATA=OUT4;  
PLOT ABSRES*P='*';  
RUN;  
QUIT;
```

SAS commands DATA OUT4 and SET OUT1 tell SAS to use content of WORK.OUT1 to create a data set called WORK.OUT4. Function ABS calculates absolute values of residuals stored in a variable called ABSRES.).

☐ Using the mouse, highlight the SAS commands above and click on the run icon.

The plot of interest is given below.



Problem: using temporary SAS data set WORK.OUT4 created above, use SAS Interactive to make a nonparametric loess smooth of the plot of absolute values of residuals versus predicted values. Use the option mean to smooth the curve.

- ☐ Save the SAS program file.
- ☐ Clear the log-window.
- ☐ Save the output window's content as follows: Choose **File > Save**
 - ☐ Choose A: **drive and type the name TUTOR1**. SAS automatically assigns the extension LST to your file. So the output file name is TUTOR1.LST.

The whole SAS Program file you have created so far is given below.

```

/*
THIS PROGRAM IS USED TO CREATE A SMALL SAS PROGRAM.
TOPICS OF INTEREST INCLUDE: DATA MANIPULATION,
PLOTTING DATA USING SAS INTERACTIVE, PLOTTING DATA USING SAS
COMMANDS, AND REGRESSION ANALYSIS.
WRITTEN BY: LAST NAME, FIRST NAME OF STUDENT.
DATE: JANUARY 2000.
*/
OPTIONS LS=79 NOCENTER NODATE NONUMBER;
TITLE 'DATA MANIPULATION USING SAS PROGRAM.';
TITLE2 'LAST NAME, FIRST NAME OF STUDENT. BTRY 602. DATE: ';
DATA TUTORIAL;
INFILE 'A:PLANT.TXT';
INPUT HEIGHT AMOUNT;
      SQHEIGHT = SQRT(HEIGHT);
      LNAMOUNT = LOG(AMOUNT);
      AMOUNT2=LNAMOUNT**2;
      OBS=_N_;
LABEL HEIGHT ='ADJUSTED PLANT HEIGHT'
AMOUNT ='AMOUNT OF NUTRIENT';
PROC PRINT DATA=TUTORIAL;
*PLOTTING THE DATA;
PROC PLOT DATA=TUTORIAL;
      PLOT HEIGHT*AMOUNT='+' SQHEIGHT*LNAMOUNT='x';

*PERFORMING REGRESSION ANALYSIS;
PROC REG DATA=TUTORIAL;
MODEL SQHEIGHT=LNAMOUNT AMOUNT2;
OUTPUT OUT=OUT1 STUDENT=R PREDICTED=P H=LEVER1 COOKD=CD1 ;

*DIAGNOSTIC PLOTS IN REGRESSION ANALYSIS;
PROC PLOT DATA=OUT1;
PLOT R*P = '+'/VREF=0;
PLOT LEVER1*OBS='+' CD1*OBS='+';
*NORMAL PROBABILITY PLOT;
PROC RANK DATA=OUT1 OUT=OUT2 NORMAL=BLOM;
VAR R;
RANKS NR3;
PROC PLOT DATA=OUT2;
PLOT R*NR3='*';

*ASSESSING EQUAL VARIANCE ASSUMPTION;
DATA OUT4;
SET OUT1;
ABSRES=ABS(R);
PROC PLOT DATA=OUT4;
PLOT ABSRES*P='*';
RUN;
QUIT;

```

Producing A Report From A SAS Output.

There are many different ways to produce a report using SAS output. We will go through one way, which assumes that you have Microsoft Word on your computer and that your computer is powerful enough to run Word and SAS at the same time. You will need to incorporate SAS output and graphs in your homework documents, so make sure to practice this part thoroughly. We suggest that you read through this section even if you do not have Word, just to get an idea of how to include SAS output in a report. It is assumed here that you are using Word 2000.

Start Microsoft Word 2000 as follows:

- ☐ **Start > Programs > Microsoft Office > Microsoft Word**

In Word 2000, write your report. For example, type the heading and introductory material describing the problem you analyzed. Discuss statistical analysis results. Suppose you would like to include a SAS regression analysis output and a loess smooth plot of residuals versus predicted values in your report.

Copy SAS regression analysis output to the Clipboard as follows:

- ☐ Open the SAS output window: **Window > Output**
- ☐ Choose **Tools > Options > Fonts**, and select font size 8 (unless it is already selected by default). This font is needed to produce a better output in Word.
- ☐ Scroll through the regression output to locate the information you want to copy in Word.
- ☐ Using the left mouse button, highlight the regression output portion of interest.
- ☐ Choose **Edit > Copy**.

Paste the regression output into your word document as follows:

- ☐ Open your Word document.
- ☐ Place the cursor where you want to paste the regression output.
- ☐ Choose **Edit > Paste**

Copying and pasting a graph in Word

For instance, suppose you want to copy in Word the loess smooth plot of residuals versus predicted values created using SAS Interactive Data analysis.

- ☐ Click on the border of the graph. This will create a thick black box around the graph.
- ☐ Choose **Edit > Copy**.
- ☐ Open the Word document.
- ☐ Place your cursor where you want to insert the graph.
- ☐ Choose **Edit > Paste**. The graph will float over the text so you need to make adjustments.
- ☐ Right-click on your graph and then choose **Edit > Close Picture**
- ☐ Save your report in Microsoft Word: **File > Save > A:\TUTOR1.DOC**
- ☐ Now you can print your report: **File > Print > Click OK**

Note: Please, make sure to exit both SAS and Word Programs before leaving the computer lab.

- ☐ Choose **File > Exit** to quit a program.

Recommended Reference

1. SAS Procedures, Version 6
2. SAS User's Guide, Version 7.1, Volume 1-6.
3. SAS Language and Procedures, Usage, Version 6
4. SAS User's Guide: Basics, Version 5
5. SAS System for Linear Models, 3rd Edition
6. SAS/STAT User's Guide, Version 6
7. Learning SAS in the Computer Lab, 2nd Edition.
8. <http://ftp.sas.com/samples/>. This web site contains interesting SAS documents.
9. <http://v8doc.sas.com/sashtml/>. This web site contains SAS online documents.

SAS TUTORIAL ENDS HERE. PAGES 38 THROUGH THE END ARE PROVIDED FOR FUTURE REFERENCE.
--

PART II. USEFUL INFORMATION ON SAS PROGRAM

Appendix 1: Data Management And Types Of SAS Data Sets

The information provided below is not required for the SAS tutorial. You should work on these SAS commands at another convenient time.

Creating Temporary SAS Data Sets

To create or read a temporary SAS data set, you usually specify only the NAME. The SAS system automatically assigns the LIBREF (library reference name) WORK to the data set, so the SAS system creates a temporary SAS data set named WORK.EXAMPLE.

```
DATA EXAMPLE;
```

```
INPUT NAME $ SEX $ AGE HEIGHT WEIGHT;
```

```
DATALINES;
```

```
Aubrey M 41 74 170
```

```
Ron M 42 68 166
```

```
Carl M 32 70 155
```

```
Antonio M 39 72 167
```

```
Deborah F 30 66 124
```

```
Jacqueline F 33 66 115
```

```
Helen F 26 64 121
```

```
David M 30 71 158
```

```
James M 53 72 175
```

```
Michael M 32 69 143
```

```
;
```

```
PROC PRINT DATA=EXAMPLE;
```

```
RUN;
```

```
QUIT;
```

Important warning: The symbol \$ must be used in the INPUT statement, after each character variable. When referring to a temporary SAS data set, you only need to specify its name. For example, the following SAS statements automatically read and print the temporary SAS data set WORK.EXAMPLE.

```
PROC PRINT DATA=EXAMPLE;
```

```
RUN;
```

Note: A WORK library is always created at the beginning of every SAS session and it is used by SAS to store temporary data sets. All SAS data sets with the LIBREF WORK are automatically erased at the end of the SAS job or session.

Creating Permanent SAS Data Sets Within A SAS Program File

A library reference name (LIBREF) for a permanent SAS data set is created using the statement LIBNAME. When creating a permanent SAS data set, you must specify both the LIBREF and the NAME of the permanent SAS data set. You must specify a LIBREF other than WORK because the SAS system reserves the library reference name "WORK" for temporary SAS data sets.

When referring to a permanent SAS data set, you must also specify both the LIBREF and the NAME of the permanent SAS data set.

For example, let us create a permanent SAS data set called SAMPLE.EXAMPLE and store it in the subdirectory C:\DATA. In this particular case, data are entered within SAS program using the CARDS statement. Three steps are needed here: first, create the subdirectory C:\DATA unless it already exists. Second, create the library reference name SAMPLE in the subdirectory C:\DATA using the statement LIBNAME. Finally, create the data set and store it in the permanent SAS data set named SAMPLE.EXAMPLE within the subdirectory C:\DATA. Details are presented in the following SAS program file:

```
LIBNAME SAMPLE 'C:\DATA';
DATA SAMPLE.EXAMPLE;
INPUT NAME $ SEX $ AGE HEIGHT WEIGHT;
DATALINES;
Aubrey      M      41      74      170
Ron         M      42      68      166
Carl        M      32      70      155
Antonio     M      39      72      167
Deborah     F      30      66      124
Jacqueline  F      33      66      115
Helen       F      26      64      121
David       M      30      71      158
James       M      53      72      175
Michael     M      32      69      143
;
PROC PRINT DATA=SAMPLE.EXAMPLE;
RUN;
PROC PLOT DATA=SAMPLE.EXAMPLE;
PLOT WEIGHT*HEIGHT;
RUN;
QUIT;
```

If you want to print the data or make a scatter plot of variable WEIGHT against HEIGHT, the SAS statements above are used.

Creating Permanent SAS Data Sets From An External File

To create a permanent SAS data set from an existing external file, you proceed as above (see I.2). However, you just need to read the data set in SAS. Suppose you want to create a permanent SAS data set from an external file named TUTOR1.TXT and stored on a diskette in drive A:. Let us assume that the file TUTOR1.TXT has three columns of data labeled respectively COUNTRY, POPUL and GNP. First, we need to create a library where to store the file. Suppose we want to create a library named SAMPLE (any other name can be used; e.g. WINTER) in the subdirectory C:\DATA.

- ❑ Using MSDOS, create subdirectory data on C: drive.
- ❑ Open the Window editor and type the following commands.

```
LIBNAME SAMPLE 'C:\DATA';  
DATA SAMPLE.EXAMPLE;  
INFILE 'A:\TUTOR1.TXT';  
INPUT HEIGHT AMOUNT;  
RUN;  
QUIT;
```

With the SAS statements above, a permanent SAS data set named EXAMPLE is created from an external file TUTOR1.TXT stored on A: drive. The created permanent SAS data set EXAMPLE is stored in a library called SAMPLE located in the subdirectory C:\DATA. This permanent SAS data set is referred to as SAMPLE.EXAMPLE.

Although a permanent SAS data set can not be read by other programs, there are several advantages to storing your data in a permanent SAS data set rather than leaving them in an external file or recreating a temporary data set in every SAS job:

- You leave reading the data to SAS; you don't have to be concerned about format, and there is no need to execute an INPUT statement each time a SAS data set is used.
- SAS automatically documents the SAS data set, and you can keep track of its content easily. Using SAS utility procedures, you can always find out which variables the data set contains, their length and formats, and other information that often is lost for undocumented files.
- No data conversion is necessary since data are stored in the form in which SAS can use them, saving computer time.
- Variable names and labels are saved with the data, so these do not need to be re-entered.

Appendix 2: Important SAS Commands For Old Homework

Appendix 2.1. Potentially Useful SAS commands For Homework #2.

Please refer to the SAS Tutorial for SAS commands needed in your analysis. For loess smooth, use SAS Interactive following instructions in SAS Tutorial as needed.

You must use the names of the variables as given in the data set description.

```
DATA HW1;  
INFILE 'A:\HOME1.TXT' FIRSTOBS=10;  
INPUT TRAP XVAR YVAR;  
XVAR2=SQRT(XVAR);  
OBS=_N_;
```

INFILE tells SAS to read data from line 10 as it is where the first data entry is located. The nine preceding lines are just used to describe the data set.

New SAS commands for computing 95% CI and 95% PI.

In simple linear regression, suppose we want a 95% confidence interval for the mean and a 95% prediction interval for a new observation of Y if the independent variable (XVAR) takes on values: 13.9, and 36.8. It is assumed that the independent variable XVAR was transformed by taking square roots. Refer to other transformations discussed in the SAS tutorial if and whenever needed.

```
DATA NEW;  
INPUT XVAR ;  
XVAR2=SQRT(XVAR);  
CARDS;  
13.9  
36.8  
;
```

We create a new data set called WORK.NEW to store values for the independent variable.
We need only to input the independent variable XVAR which is actually used in the regression model.

We want to transform the data for the predictor variable (used to compute confidence and prediction intervals) by taking their square roots as we did for the original data.

This command tells SAS that values of the independent variable are immediately following. It is a holdover from the days when computer cards were used.

A semi-colon after the last data entry tells SAS that this is the end of the data values.

```
DATA BOTH;  
SET NEW HW1;
```

We use statement SET to append the SAS data set WORK.NEW at the top of the original SAS data set WORK.HW1, thus making a new SAS data set called WORK.BOTH. This sequence (first WORK.NEW then WORK.HW1) is advised to have the confidence and prediction intervals at the top of the SAS output of interest. In this case, the first two observations of the CI's and PI's output are the requested intervals for respectively XVAR = 13.9 and XVAR = 36.8.

```
PROC REG DATA=BOTH;
```

The first two entries in data WORK.BOTH are not used in parameters' estimation as they do not have Y values. They are used to compute confidence and prediction intervals.

```
MODEL YVAR=XVAR2/CLM CLI;
```

```
OUTPUT OUT=OUT1 PREDICTED=P STUDENT=R;
```

```
DATA OUT2;
```

```
SET OUT1;
```

```
ABSRES=ABS(R);
```

Data WORK.OUT2 is created using WORK.OUT1. The function ABS calculates absolute values. ABSRES stores the absolute values of the residuals (R).

Option CLM asks SAS to compute 95% confidence intervals for the fitted values. Option CLI asks SAS to compute 95% prediction intervals. SAS computes the prediction interval for every point in the combined data set.

Appendix 2.2. Potentially Useful SAS commands For Homework #3.

Please refer to the SAS Tutorial and Homework #2 for SAS commands needed in your analysis. For loess smooth, use SAS Interactive following instructions in SAS tutorial as needed.

You must use the names of the variables as given in the data set description.

If you want to transform your data, you should do so in the DATA step. Below are examples of some transformations of variables.

```
DATA HW3;
```

```
INFILE 'A:\home3.txt';
```

```
INPUT COL1 COL2;
```

```
VARY2 = SQRT(COL1);
```

VARY2 gives the square root of COL1

```
VARY3 = LOG(COL2);
```

VARY3 gives the natural logarithm (base e) of COL2

```
VARY4 = LOG10(COL2);
```

VARY4 gives the logarithm (base 10) of COL2

```
VARY5 = 1/(SQRT(COL1));
```

VARY5 gives the inverse of the square root of COL1

```
PROC REG DATA=HW3;
```

```
MODEL Y = X/NOINT;
```

Option NOINT tells SAS not to include the intercept in the regression model. So SAS performs regression through the origin.

```
OUTPUT OUT=OUT1 PREDICTED=P STUDENT=R;
```

```
DATA OUT2;
```

```
SET OUT1;
```

```
ABSRES=ABS(R);
```

Appendix 2.3. Potentially Useful SAS commands for Homework #4.

Please, refer to the SAS Tutorial and previous assignments for SAS commands or parts of SAS Interactive needed in your analysis. Use rationally SAS commands provided herein as they are not necessarily all needed.

```
DATA HW4;
```

```
INFILE 'A:NAME.TXT';
```

```
INPUT IDSTATE STATE $ TAX INCOME ROAD DLIC FUEL;
```

The symbol \$ tells SAS that variable STATE is not numeric.

```
OBS=_N_;
```

IF OBS=38 THEN DELETE;

Statements above instruct SAS to delete observation #38. This is a handy way of removing influential data points. **(In this example, we show how we would remove data point #38 if it were influential. So, use the appropriate observation # in your SAS program.)**

PROC CORR DATA=HW4; computes the correlation coefficients among variables

VAR TAX INCOME ROAD DLIC FUEL;

PROC REG DATA=HW4;

MODEL FUEL = TAX INCOME ROAD DLIC / SS1 SS2 VIF TOL PARTIAL;

OUTPUT OUT=OUT1 STUDENT=R PREDICTED=P COOKD=CD1 H=LEVER1;

SS1 computes the sequential (extra) SSR

SS2 computes the partial SSR

VIF computes the variance inflation factors

TOL computes the tolerance

PARTIAL prints all partial regression plots

For confidence intervals or prediction intervals, make sure to input all predictor variables of interest. (See example below).

DATA MYNAME;

INPUT STATE \$ TAX INCOME ROAD DLIC;

CARDS;

STATE#1 9.00 5200 3350 0.48

STATE#1 8.00 4690 602 0.648

;

DATA ALL;

SET MYNAME HW4;

PROC REG DATA= ALL;

MODEL FUEL = TAX INCOME ROAD DLIC / CLM ALPHA=0.01;

CLM ALPHA = 0.01 tells SAS to compute 99% confidence intervals. (For 90%CI, use ALPHA = 0.10).

Appendix 2.4. Potentially Useful SAS commands for Homework #6.

We have two options when performing regression with qualitative and quantitative predictors.

Option 1: Use of PROC GLM.

With PROC GLM, indicator variables are created with the CLASS statement.

OPTIONS LS=79 NONUMBER NODATE NOCENTER;

DATA TONY;

INFILE 'A:\SALARY.TXT' FIRSTOBS=12;

PROC GLM DATA=TONY;

CLASS EDUCN JOB;

MODEL SALARY = EXPERN EDUCN JOB EDUCN*JOB/SS1 SS3 SOLUTION;

Notice that we did not create indicator variable for EDUCN. The indicator variables are created by statement CLASS EDUCN. Since variable EDUCN has 3 categories, CLASS EDUCN creates 3 indicator variables (instead of just 2 created in the INPUT statement for regression). CLASS does also create two indicator variables for the two categories of JOB.

EDUCN tells SAS to use the 3 categorical variables created whereas JOB tells SAS the two indicator variables created in statement CLASS.

EDUCN*JOB creates interaction terms between JOB and each of the three indicator variables. (EDUCN*EXPERN would create interaction terms between EDUCN and EXPERN.)

Presence of 3 indicator variables along with an intercept in the model lead to a singular X'X matrix (similar problem with JOB indicator variables). In other words, this matrix does not have an inverse since the sum of the columns of these indicator variables is

equal to the column of ones for the intercept. To handle this problem SAS uses the first two indicator variables and sets the last one to zero.

Option SS1 calculates extra or sequential sum of squares. Option SS3 calculates partial sum of squares. Option SOLUTION tells SAS to compute regression coefficients of the model.

```
OUTPUT OUT=ONE RSTUDENT=R PREDICTED = P;
PROC PLOT DATA=ONE;
PLOT R*P=GROUP;
RUN;
```

Note: PROC GLM offers a greater flexibility especially in creating indicator variables. This is especially true when the number of indicator variables to be created is high.

Option 2: Use of PROC REG.

With PROC REG, we need to create all indicator variables in the INPUT statement.

```
OPTIONS LS=79 NONUMBER NODATE NOCENTER;
DATA TONY;
INFILE 'A:\COMPUTER.TXT' FIRSTOBS=18;
INPUT IDNUM SALARY EXPERN EDUCN JOB;
IF EDUCN=1 AND JOB=0 THEN GROUP=1;
IF EDUCN=2 AND JOB=0 THEN GROUP=2;
IF EDUCN=3 AND JOB=0 THEN GROUP=3;
IF EDUCN=1 AND JOB=1 THEN GROUP=4;
IF EDUCN=2 AND JOB=1 THEN GROUP=5;
IF EDUCN=3 AND JOB=1 THEN GROUP=6;
```

Logic operators IF and THEN are used to create a new categorical variable named GROUP. This variable will be used to plot the data.

```
OBS=_N_;
EDU1=(EDUCN=1);
EDU2=(EDUCN=2);
```

This is what is called a logic operation. EDU1=1 if the statement in the bracket is correct, and 0 if it is false. We only need two dummy variables here because the number of categories = 3. Values for EDUCN = 3 are set to EDU1 = EDU2 = 0. Note that single quotes would be needed if the categories for EDUCN were categorical. For example if we had categories A, B, and C, then we would create dummy variables as follows:

```
EDU1=(EDUCN='A');
EDU2=(EDUCN='B');
```

```
EDJOB14=EDU1*JOB;
EDJOB24=EDU2*JOB;
```

The interaction terms between EDUCN and JOB are formed by multiplication. Other interaction terms are created in a similar way whenever needed.

```
EXPEDU1=EDU1*EXPERN;
EXPEDU2=EDU2*EXPERN;
```

These are interaction terms between EDUCN and EXPERN.

```
PROC REG DATA=TONY;
MODEL SALARY = EXPERN EDU1 EDU2 JOB EDJOB14 EDJOB24/SS1 SS2;
```

The test of the interaction terms in the regression model will enable us to assess whether the three regression slopes are the same or not. Statistics of interest in multiple regression can be saved in SAS as before using the OUTPUT statement.

```
PROC PLOT DATA=TONY;
PLOT SALARY*EXPERN = GROUP;
```

Values of GROUP are used as plotting symbols on the scatter plot of SALARY against EXPERN.

```
RUN;
```

SAS Commands for Variable Selection:

Please, use the correct variable names in your SAS program (do not use the names used in this illustration: YVAR, XVAR1, etc...). Note that usually we would not use all three techniques. PROC RSQUARE is the preferred method. (Not all of the needed SAS commands are provided. Refer to previous homework assignments whenever necessary.) The example below is based on a hypothetical case involving nine independent variables and one dependent variable. Extrapolation to more or less independent variables is straightforward.

```
OPTIONS LS=79 NONUMBER NOCENTER NODATE;  
DATA HOME6;  
INFILE 'A:COMPUTER.TXT' FIRSTOBS=18;  
INPUT OBS YVAR XVAR1 XVAR2 XVAR3 XVAR4 XVAR5 XVAR6 XVAR7  
        XVAR8 XVAR9;
```

```
PROC RSQUARE OUTEST=EST SELECT=1;
```

PROC RSQUARE is the SAS procedure for doing all subsets regression. We generally need to do 2 runs of PROC RSQUARE. The first run is to obtain the data needed for the plot of R^2 versus number of parameters. This plot helps determine the number of variables needed in the model.

OUTEST=EST saves the value of R^2 in a SAS variable called `_RSQ_`, and the number of parameters in a SAS variable called `_P_`.

SELECT=K tells PROC RSQUARE to determine only the models with the K highest R^2 for each number of parameters.

```
MODEL YVAR=XVAR1 XVAR2 XVAR3 XVAR4 XVAR5 XVAR6 XVAR7  
        XVAR8 XVAR9/SELECT = 1;
```

The model statement is similar to the model statement in PROC REG. The option SELECT=1 must be repeated.

```
PROC PLOT DATA=EST;  
PLOT_RSQ_*_P_='*';
```

We plot R^2 versus number of parameters. Generally, we would stop here, and do the next step in a separate run, after deciding the number of variables needed in the model.

```
PROC RSQUARE DATA=HOME6 SELECT=2;
```

We run PROC RSQAURE a second time. This time we do not save R^2 or the number of parameters. However, we compute the best 2 models for each number of parameters. We seldom want to consider more than a few models as the cost of computing more models goes up rapidly.

```
MODEL YVAR= XVAR1 XVAR2 XVAR3 XVAR4 XVAR5 XVAR6 XVAR7  
        XVAR8 XVAR9/ SELECT = 2 STOP=6;
```

The option STOP=6 tells SAS not to compute models which have more than 6 variables (not parameters). Generally, we would look at the plot of R^2 versus number of parameters to determine how many variables we want to include in the model, and perhaps go one variable higher.

PROC STEPWISE; PROC STEPWISE does stepwise regression.

```
MODEL YVAR= XVAR1 XVAR2 XVAR3 XVAR4 XVAR5 XVAR6 XVAR7 XVAR8 XVAR9 / FORWARD  
BACKWARD STEPWISE SLE=0.15 SLS=0.10;
```

The options FORWARD, BACKWARD, and STEPWISE tell SAS to use the forward, backward, or stepwise selection method. (Usually, we would select only one of these methods.). FORWARD model selection method begins with no variable and adds one variable at a time until no variable more variable meets the selection criterion.

BACKWARD model selection method begins with all variables in the model and removes one variable at a time until all remaining variables are statistically significant, with respect to the selection criterion. STEPWISE model selection is a combination of the previous two selection methods (FORWARD and BACKWARD).

SLE specifies the significance level for entry into the model used for FORWARD and STEPWISE methods. The defaults are 0.50 for FORWARD and 0.15 for STEPWISE. SLS specifies the significance level for staying in the model used for BACKWARD and STEPWISE methods. The defaults are 0.10 for BACKWARD and 0.15 for STEPWISE.

REMARKS:

We can also use PROC REG with the above selection methods as follows to get similar results.

```
PROC REG;  
MODEL YVAR= XVAR1 XVAR2 XVAR3 XVAR4 XVAR5 XVAR6 XVAR7 XVAR8 XVAR9 /  
SELECTION=RSQUARE CP BEST=5;
```

These commands are used for the RSQUARE selection method. Option CP gives the C_p statistic. The best 5 models are selected at each number of parameters.

```
PROC REG;  
MODEL YVAR= XVAR1 XVAR2 XVAR3 XVAR4 XVAR5 XVAR6 XVAR7 XVAR8 XVAR9 /  
METHOD=FORWARD;
```

These commands are used for the FORWARD selection method.

```
PROC REG;  
MODEL YVAR= YVAR= XVAR1 XVAR2 XVAR3 XVAR4 XVAR5 XVAR6 XVAR7 XVAR8 XVAR9 /  
METHOD= BACKWARD;
```

These commands are used for the BACKWARD selection method.

```
PROC REG;  
MODEL YVAR= XVAR1 XVAR2 XVAR3 XVAR4 XVAR5 XVAR6 XVAR7 XVAR8 XVAR9 / METHOD=  
STEPWISE;
```

These commands are used for the true STEPWISE selection method.

NOTE: At this point you have mastered SAS Programming to some extent. SAS commands not provided here, but covered in previous homework assignments should be included as it becomes necessary. The new SAS commands above should be used as a complement to other SAS commands already covered. Incorporate intelligently these new SAS commands in your SAS program file whenever necessary.

Appendix 2.5. Potentially Useful SAS commands For Homework #7.

SIMPLE LOGISTIC REGRESSION, GROUPED DATA

Other commands were provided in the lecture notes and lab recitation handouts. So please check these additional sources of information if necessary. If the number of successes (NSUCCESS) and failures (NFAILURE) are given, then the total number trials $NTRIALS = NSUCCESS + NFAILURE$ must be computed in the INPUT statement. In the example below, one is given the number of successes (NSUCCESS) and the number of trials (NTRIALS) for each level of predictor variable WIDTH.

```
DATA CRABS;  
INFILE 'A:BOB.TXT'  
INPUT WIDTH NTRIALS NSUCCESS;
```

```
PROC LOGISTIC DATA=CRABS;  
MODEL NSUCCESS/NTRIALS = WIDTH / LACKFIT COVB;  
OUTPUT OUT=OUT1 P=PRED LOWER=L95 UPPER=U95;  
RUN;
```

PROC LOGISTIC is used to fit a logistic regression model the data. LACKFIT option requests lack of fit test based on Hosmer and Lemeshow goodness-of-fit test statistic.

Option COVB computes the covariance matrix of the logistic regression coefficients' estimates.

Do not use the option ALPHA as in ordinary regression, as computation of simultaneous confidence intervals is not available for probabilities with the current SAS Release 8.1.

Option P, LOWER, UPPER give respectively predicted probabilities and 95% lower and upper confidence intervals for population probabilities.

```
PROC PLOT DATA=OUT1 HPERCENT=50 VPERCENT=50;  
PLOT P*WIDTH='P' LOWER*WIDTH='*' UPPER*WIDTH='*' /OVERLAY;  
RUN;
```

Option OVERLAY the predicted along with the lower and 95 % confidence intervals on the same graph.

```
PROC PRINT DATA=OUT1;  
RUN;  
DATA PREDs;  
INPUT WIDTH;  
DATALINES;  
25  
27  
;
```

```
DATA ALL;  
SET PREDs CRABS;  
RUN;
```

```
PROC LOGISTIC DATA=ALL;  
MODEL NSUCCESS/NTRIALS = WIDTH / CLM COVB;  
RUN;
```

By default, option CLM is used to compute 95% confidence intervals for probabilities. No adjustments of simultaneous confidence intervals are feasible unlike in ordinary regression. So, one has to do computations manually or do extra programming to get simultaneous CI's for probabilities.

Option COVB is used to compute the covariance matrix of estimated regression coefficients.

SIMPLE LOGISTIC REGRESSION, UNGROUPED DATA

```
OPTIONS LS=79 NONUMBER NODATE NOCENTER;  
DATA ONE; INFILE 'A:HWS.TXT';  
INPUT OBS ESR FIBRIN;  
PROC LOGISTIC DATA=ONE DESCENDING;  
MODEL ESR=FIBRIN;
```

Option DESCENDING is needed because SAS fits by default $Y=0$ or a smaller value of Y as the success. We need this option to reverse the order of the value of the response so that the success event coded $Y=1$ is fitted.

```
RUN;
```

MULTIPLE LOGISTIC REGRESSION, UNGROUPED DATA

The overall statements used above, for simple logistic regression, are valid as in multiple logistic regression analysis. Just include additional predictors of interest under the model statement.

```
PROC LOGISTIC DATA=ONE DESCENDING;  
MODEL Y=XVAR1 XVAR2 XVA3 XVAR4;  
RUN;
```

MULTIPLE LOGISTIC REGRESSION, GROUPED DATA

SAS commands used for simple logistic regression (grouped data) are valid here as well. Just incorporate as many predictor variables as need in the model statement.

```
PROC LOGISTIC DATA=ALL;  
MODEL NSUCCESS/NTRIALS = XVAR1 XVAR2 XVAR3 XVAR4 / LACKFIT RSQ;
```

Option RSQ requests computation another measure of model's goodness fit of test as is the case with R^2 value in multiple linear regression analysis. However these two statistics do not have the same interpretation as in regression analysis.

```
RUN;
```

MULTIPLE LOGISTIC REGRESSION WITH CATEGORICAL PREDICTORS

In the example below one is dealing with multiple logistic regression with all predictors being categorical. One also can use these commands for grouped data. Residuals are plotted against predicted values to assess whether the model fits well the data as in regression analysis.

```
PROC LOGISTIC DATA=ASGOOD;  
CLASS YEAR ED SOUTH/PARAM=GLM;  
MODEL ASGOOD/TOT= YEAR|ED|SOUTH @2/CLPARM=BOTH LACKFIT;  
OUTPUT OUT=NEWONE(KEEP=PRED LOWER UPPER CHI DEV) P=PRED U=UPPER L=LOWER  
RESCHI=CHI RESDEV=DEV;
```

CLASS statement creates indicator variables in GLM for categorical predictor variables: YEAR ED and SOUTH.

PARAM=GLM option requests that the category with the largest code gets the value zero, the same as for indicator variables that are created with PROC GLM.

Notation YEAR|ED|SOUTH @2 in the model statement is a handy way of creating all main factor effect and all second order interaction terms among these three predictors YEAR ED and SOUTH.

So @2 indicates main effects and second order interactions. For main effects, second order and third order interactions, one would use @3.

Option CLPARM=BOTH calculates 95% confidence intervals for the odds ratio of the event of interest (success) for each predictor variable in the model, using both the Wald test statistic and likelihood ratio statistic.

Option OUT=NEWONE(KEEP=PRED LOWER UPPER CHI DEV) creates SAS data named NEWONE that stores variables:

PRED = P, predicted probabilities

LOWER=L, lower 95% limit for $\pi(x)$

UPPER=P, upper 95% limit for $\pi(x)$

RESCHI=CHI, Pearson chi-square residuals

RESDEV=DEV, deviance residuals

KEEP option is used to avoid having all variables and statistics in data set NEWONE.

RUN;

MULTIPLE LOGISTIC REGRESSION WITH INDIVIDUAL RESPONSE

The response variable Y is binary (Y=1, or Y=0). Suppose one has four predictors X1, X2, X3, and X4. We simply need to add more predictors in the model statement below. Options under logistic regression with categorical response are valid when predictors are quantitative.

OPTIONS LS=79 PS=120 NONUMBER NODATE NOCENTER;

DATA ONE;

TITLE 'LOGISTIC REGRESSION FOR INDIVIDUAL DATA';

INPUT OBS Y X1 X2 X3 X4;

PROC LOGISTIC DATA=ONE DESCENDING;

MODEL Y= X1 X2 X3 X4/LACKFIT;

RUN;

Appendix 2.6. Potentially Useful SAS commands For Homework #8.

The SAS commands below are from the class handout. So make sure to edit the file by using the appropriate variables names.

DATA ONE;

INFILE 'A:JOHN.TXT';

INPUT OBSNUM TREAT \$ LENGTH;

IF TREAT='A' THEN SOAKTIME=12;

IF TREAT='B' THEN SOAKTIME=18;

IF TREAT='C' THEN SOAKTIME=24;

IF TREAT='D' THEN SOAKTIME=30;

PROC GLM DATA=ONE;

CLASS TREAT;

MODEL LENGTH=TREAT;

MEANS TREAT;

MEANS TREAT/TUKEY CLDIFF ALPHA = 0.05;

RUN;

MEANS statement calculates means for levels of treatment TREAT. Multiple comparisons used are based on Tukey's method. The Option CLDIFF calculates 95% simultaneous confidence intervals between two treatment means.

* PERFORMING LACK OF FIT TEST;

```
PROC GLM DATA=ONE;  
CLASS TREAT;  
MODEL LENGTH=SOAKTIME SOAKTIME*SOAKTIME TREAT/SS1;  
RUN;
```

In the MODEL statements above: SOAKTIME is used as the linear term,
SOAKTIME*SOAKTIME is used as the quadratic term and
TREAT is used for assessing lack-of-fit.

```
PROC GLM DATA=ONE NOPRINT;  
CLASS TREAT;  
MODEL LENGTH=TREAT;  
OUTPUT OUT=OUT1 RSTUDENT=R P=P;  
RUN;
```

Option NOPRINT instructs SAS not to print output from PROC GLM since we already have this information from previous SAS commands. We are just interested in residuals needed for assessing assumptions of the linear model.

* ASSESSING ASSUMPTIONS;

```
PROC RANK DATA=OUT1 OUT=OUT3 NORMAL=BLOM;  
VAR R;  
RANKS NR;
```

```
DATA ALL;  
SET OUT3;  
ABSRES=ABS(R);
```

```
PROC PLOT DATA=ALL VPERCENT=50;  
PLOT ABSRES*P='*' R*P='*' R*NR='*';  
RUN;  
QUIT;
```

Testing contrasts using SAS.

In the example below, we assume that there are 6 treatments of interest (A, B, C, D, E and F). First, we would like to compute treatment means. Second, we would like to test the null hypothesis $H_0: L = 0$ versus $H_A: L \neq 0$. Suppose that there are three contrasts (L_1 , L_2 , and L_3) of interest in this experiment.

$$L_1 = \mu_1 - \mu_2 \Leftrightarrow L_1 = \mu_1 - \mu_2 + 0\mu_3 + 0\mu_4 + 0\mu_5 + 0\mu_6$$

$$L_2 = \frac{1}{2}(\mu_1 + \mu_2) - \frac{1}{3}(\mu_4 + \mu_5 + \mu_6) \Leftrightarrow L_2 = \frac{1}{2}(\mu_1 + \mu_2) + 0\mu_3 - \frac{1}{3}(\mu_4 + \mu_5 + \mu_6)$$

$$L_3 = \frac{1}{3}(\mu_1 + \mu_4 + \mu_5) - \frac{1}{3}(\mu_2 + \mu_3 + \mu_6) \Leftrightarrow L_3 = \frac{1}{3}\mu_1 - \frac{1}{3}\mu_2 - \frac{1}{3}\mu_3 + \frac{1}{3}\mu_4 + \frac{1}{3}\mu_5 - \frac{1}{3}\mu_6$$

```
* SAS COMMANDS FOR COMPUTING TREATMENT MEANS;  
PROC GLM;  
CLASS TREAT;  
MODEL Y = TREAT/SOLUTION;  
MEANS TREAT;  
LSMEANS TREAT;
```

Statements MEANS and LSMEANS give respectively weighted and unweighted treatment means. LSMEANS is preferred in unbalanced factorial designs.

```

* SAS COMMANDS FOR TESTING SIGNIFICANCE OF A CONTRAST USING T-TEST;
PROC GLM;
CLASS TREAT;
MODEL Y = TREAT;
ESTIMATE 'L1 TEST' TREAT  1 -1 0 0 0 0; /* L1 coefficients */
ESTIMATE 'L2 TEST' TREAT  3 3 0 -2 -2 -2/DIVISOR=6; /* 1/6 of L2 coefficients */
ESTIMATE 'L3 TEST' TREAT  1 -1 -1 1 1 -1/DIVISOR=3; /* 1/3 of L3 coefficients */

```

```

* SAS COMMANDS FOR TESTING SIGNIFICANCE OF A CONTRAST USING F-TEST;
PROC GLM;
CLASS TREAT;
MODEL Y = TREAT;
CONTRAST 'L1 TEST' TREAT  1 -1 0 0 0 0; /* L1 coefficients */
CONTRAST 'L2 TEST' TREAT  3 3 0 -2 -2 -2; /* L2 coefficients*6 */
CONTRAST 'L3 TEST' TREAT  1 -1 -1 1 1 -1; /* L3 coefficients*3 */

```

The ESTIMATE statement computes the estimated value of a contrast, its standard error and performs a two-sided T-TEST of the hypothesis that a contrast is equal to zero. The SAS option DIVISOR is used to avoid rounding errors in the values of coefficients.

The CONTRAST statement gives results similar to the ESTIMATE statement. However, it computes F-values instead of T-values. In other words, it is based on the F-TEST statistic. Note that the option DIVISOR is not used here.

Note:

For other SAS commands, refer to previous assignments.

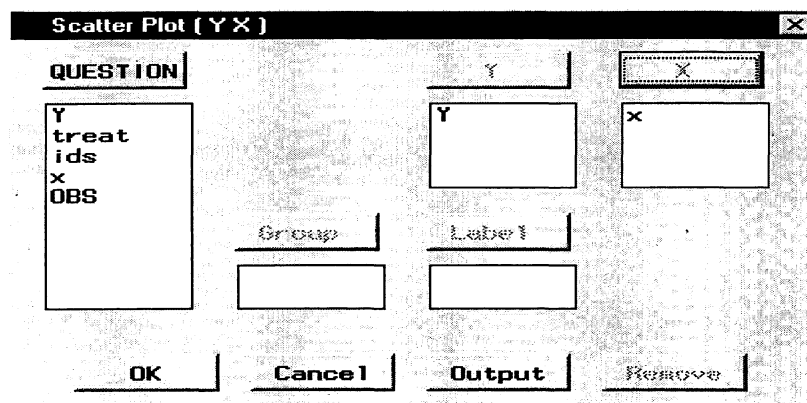
Appendix 2.7. Potentially Useful SAS commands For Homework #9.

I. Plotting symbols on scatter plots with SAS Interactive.

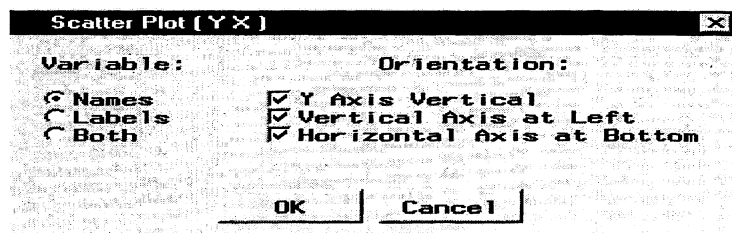
Suppose we want to plot the response variable **Y** against the independent variable **X** using variable **treat** as the plotting symbol variable.

To label data points on a SAS Interactive scatter plot, proceed as follows:

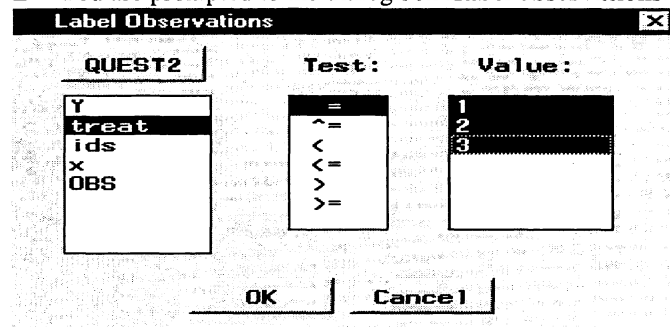
- ☐ Open the appropriate temporary SAS data in SAS interactive.
- ☐ Select Analyze > Choose Scatter plot (Y X).
- ☐ You are prompted to the dialog box below.



- ☐ In the dialog box above, choose the dependent variable **Y** and the independent variable **X**.
- ☐ Select the plotting symbol variable **treat** then click on the **label** tab.
- ☐ Click on the **Output** tab.



- ☐ Select **Labels** in the dialog box above.
- ☐ Click OK twice.
- ☐ Choose Edit > Observations > Label in Plot.
- ☐ You are prompted to the dialog box **Label observations**.



- ☐ Select **treat**, the plotting symbols variable of interest.
- ☐ Under **Value**, hold down the shift key and use your left mouse button to select all values (1, 2, 3) of this plotting variable.
- ☐ Click OK. Your plotting symbols should now appear on the scatter plot along with data points.
- ☐ To remove data points on your graph and keep labels only, click on the lower left arrow of the box around your graph and deselect observations.

II. Important SAS Commands.

For other SAS commands, refer to previous assignments.

1. ANCOVA SAS commands.

Assessing the significance of the interaction.

```
PROC GLM;
CLASS TREAT;
MODEL Y = TREAT X TREAT*X/SOLUTION;
```

Performing analysis of covariance.

```
PROC GLM;
CLASS TREAT;
MODEL Y = X TREAT/SOLUTION;
LSMEANS TREAT/STDERR PDIF;
/* SOLUTION option gives the solution to the normal equations. LSMEANS gives adjusted treatment means.
STDERR gives standard errors of the adjusted treatment means. PDIF gives p-values for the test of equality of
pairs of means. */
```

2. TWO-WAY ANOVA SAS commands.

Suppose we have two factors of interest: A (with levels a1 and a2) and B (with 3 levels b1, b2 and b3). Suppose we want to fit a linear model $Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk}$

where $\mu_{ij} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij}$

```
PROC GLM;
CLASS A B;
MODEL Y = A B A*B/SS3;
```

It is advisable to request type III sum of squares (SS3) as it is valid for balanced or unbalanced designs with PROC GLM.

Multiple comparisons of means for balanced designs.

```
PROC GLM;
CLASS A B;
MODEL Y = A B A*B/SS3;
MEANS A B AB/TUKEY;
```

Multiple comparisons of means are easily extended to factorial designs. Just choose the multiple comparison method of interest.

Testing contrasts comparisons of means.

Three steps are required:

- First write down each mean in the contrast in terms of model parameters
- Using expressions of means in (a) to derive the contrast
- Order the parameters of the contrast.

Use the coefficient of the contrast in (c) in ESTIMATE statement.

Suppose we have a significant interaction and we want to test the contrast L_1 defined below.

$L_1 = \mu_{11} - \mu_{13}$. Following the 3 steps (a) through (c) above we have:

$$\mu_{11} = \mu + \alpha_1 + \beta_1 + (\alpha\beta)_{11}$$

$$\mu_{13} = \mu + \alpha_1 + \beta_3 + (\alpha\beta)_{13}$$

The contrast becomes L_1 :

$$\mu_{11} - \mu_{13} = \mu + \alpha_1 + \beta_1 + (\alpha\beta)_{11} - \mu - \alpha_1 - \beta_3 - (\alpha\beta)_{13} = \beta_1 + (\alpha\beta)_{11} - \beta_3 - (\alpha\beta)_{13}$$

Order the parameters in the contrast above to get:

$$L_1 = 1\beta_1 + 0\beta_2 - 1\beta_3 + 1(\alpha\beta)_{11} + 0(\alpha\beta)_{12} - 1(\alpha\beta)_{13} + 0(\alpha\beta)_{21} + 0(\alpha\beta)_{22} + 0(\alpha\beta)_{23}$$

```
PROC GLM;
CLASS A B;
MODEL Y = A B A*B/SS3;
LSMEANS A B A*B;
ESTIMATE 'TEST OF L1' B 1 0 -1 AB 1 0 -1 0 0 0;
```

Plotting treatment means for assessing interaction.

```
PROC SORT;
BY A B;
/* We first need to sort the data by factors A and B */
PROC MEANS MEAN;
BY A B;
VAR Y;
OUTPUT OUT=OUT1 MEAN=MEANY;
/* Means of the response variable Y are stored in a variable with the name MEANY */
PROC PLOT DATA=OUT1;
PLOT MEANY*A=B;
```

Appendix 2.8. Potentially Useful SAS commands For Homework #10.

For fixed effect factors, use PROC GLM. It is your responsibility to determine whether to use fixed effect, random effect or mixed effect factors. The SAS commands below are simply comprehensive and should be used intelligently. For contrasts, it is advisable to use statement LSMEANS. Other important SAS commands are provided in Lecture Notes and in previous assignments.

Case 1. Both A and B are random factors.

a. Using PROC GLM (balanced designs only)

```
PROC GLM;  
CLASS A B;  
MODEL Y= A B A*B;  
RANDOM A B A*B/TEST;
```

These SAS commands give the ANOVA table and expected mean squares. F-tests are correct for balanced design. PROC GLM does not give BLUP'S (best linear unbiased predictors). RANDOM statement tells SAS that A, B, and A*B are random. The TEST statement instructs SAS to use the appropriate MS at the denominator when performing an F-test statistic.

b. Using PROC MIXED (both balanced and unbalanced designs)

```
PROC MIXED;  
CLASS A B;  
MODEL Y=;  
RANDOM A B A*B;
```

No factor is given in the model statement because the two factors are random in this example. RANDOM statement tells SAS that A, B, and A*B are random.

Case 2. Factor A is fixed and Factor B is random.

a. Using PROC GLM (balanced data only)

```
PROC GLM;  
CLASS A B;  
MODEL Y= A B A*B;  
RANDOM B A*B;
```

b. Using PROC MIXED (both balanced and unbalanced designs)

```
PROC MIXED;  
CLASS A B;  
MODEL Y = A;      MODEL statement tells SAS that factor A is fixed.  
RANDOM B A*B;      RANDOM statement tells SAS that B and A*B are random.
```

Appendix 2.9. Potentially Useful SAS commands For Homework #11.

I. Important SAS Commands.

For other SAS commands, refer to lecture notes and previous assignments.

1. Using PROC GLM for Split Plot Designs.

Here we assume that the main plot factor A is fixed. We also assume that the subplot factor B and the blocking factor BLOCK are random. We finally assume that the experimental structure on the main plot is a randomized complete block design.

```
PROC GLM;  
CLASS A B BLOCK;  
MODEL Y = BLOCK A BLOCK*A B A*B;  
RANDOM BLOCK BLOCK*A B A*B/TEST;
```

Option TEST under RANDOM statement instructs SAS to use the appropriate MS's in performing F-test statistics.

One gets similar results with the following SAS commands.

```
PROC GLM;  
CLASS A B BLOCK;  
MODEL Y = BLOCK A BLOCK*A B A*B;  
RANDOM BLOCK BLOCK*A B A*B;  
TEST H=A E=A*BLOCK;
```

Statement TEST H = A E=A*BLOCK indicates that the effect of the factor A is tested using the interaction term between Factor A and Factor Block (E=A*BLOCK) as the error term in the denominator of an F-test statistic. The subplot factor B and the interaction A*B are tested using the model's mean square error by default.

2. Using PROC MIXED for Split Plot Designs.

Here we assume that the main plot factor A is fixed. We also assume that the subplot factor B and the blocking factor BLOCK are random. We finally assume that the experimental structure on the main plot is a randomized complete block design.

```
PROC MIXED;  
CLASS A B BLOCK;  
MODEL Y = A;  
RANDOM BLOCK BLOCK*A B A*B;
```

The MODEL statement contains only fixed effects
RANDOM statement is used to define random terms

3. Using PROC GLM for Latin Squares Designs.

The basic structure is similar to PROC GLM for split plot designs. Suppose we have 2 random factors A and B and a fixed factor C.

```
PROC GLM;  
CLASS A B C;  
MODEL Y = A B C;  
RANDOM A B;
```


4. Using PROC MIXED for Latin Squares Designs.

The basic structure is similar to PROC MIXED for split plot designs. Suppose we have 2 random effects factors A and B and one fixed effects factor C.

```
PROC MIXED;  
CLASS A B C;  
MODEL Y = C;  
RANDOM A B;
```

The MODEL statement contains only fixed effects factors
RANDOM statement is used to define random effects factors terms

Appendix 3: Diagram Showing SAS Program's Potential Analyses

